Haaga-Helia
ammattikorkeakoulu Oy

Tietovarastoratkaisun valintakriteeristö

Tuula Erkinheimo

Opinnäytetyö

Tietojärjestelmäosaamisen koulutusohjelma

2015

Tiivistelmä

Haaga-Helia ammattikorkeakoulu Oy

Tekijä
Tuula Erkinheimo

Koulutusohjelma
Tietojärjestelmäosaamisen koulutusohjelma

Opinnäytetyön otsikko
Tietovarastoratkaisun valintakriteeristö

Sivumäärä
47

Tutkimuksessa on tuotettu tietoa, jonka avulla on tiivistetty tietovarastoratkaisun valinnassa huomioon otettavat seikat kriteeristöksi. Kriteeristö on tarkoitettu käytännön apuvälineeksi tietovarastoratkaisujen valintatilanteissa. Tutkimuksessa on etsitty kriteeristölle teoreettisia perusteluja sekä kartoitettu erityisesti uusia tietovarastovaihtoehtoja joita on luotu Big Data –lähestymistavan myötä.

Tiedon käsittelyn tarpeet vaihtelevat saatavuuden, yksilöimisen/yksilöimättömyyden sekä kirjoitus-/lukuoperaatioiden painottumisen perusteella. Big Data ilmiönä on tuonut uudenlaisen lähestymistavan tiedon varastointiin ja tuonut uusia välineitä laajojen, vaihtelevien ja monimuotoisten tietomassojen käsittelyyn. Erityyppisillä tietovarastoratkaisuilla on erilaisia ominaisuuksia, jotka tukevat erilaisia tiedon käsittelytarpeita.

Tietomallit ovat kehittyneet ajan myötä. Nykyään käytetyin tietomalli on relaatiomalli. Erilaisia tietomalleja ovat mm. hierarkkinen tietomalli, verkkotietomalli, relaatiomalli, oliomalli sekä tietomallit, joita NoSQL- tietovarastoissa käytetään. NoSQL – tietovarastotyyppejä on kehitetty 2000-luvulla vastaamaan Big Data -ilmiön ja pilvipalveluiden tarpeita. NoSQL – termillä viitataan tietovarastoihin, joiden käsittelyyn SQL-kieli ei ole ainoa vaihtoehto. NoSQL-tietovarastojen tyypillisiä ominaisuuksia ovat tiedon skaalautuvuus horisontaalisesti useille palvelimille, relaatiotietokantojen tapahtumankäsittelyn eheys- ja turvallisuusominaisuuksista tinkiminen suorituskyvyn ja laajennettavuuden vuoksi sekä relaatiotietokantoja parempi tuki ei-strukturoidulle tiedolle. NoSQL-tietovarastot ovat usein skeemattomia tai niissä on joustava skeema. NoSQL-tietovarastoja voidaan luokitella tietomallien mukaan. Eräs yleisesti käytetty luokittelu on jako avain-arvo-varastoihin, sarakeperhevarastoihin, dokumenttivarastoihin ja verkko- l. graafitietovarastoihin.

Organisaation tulee harkita tarvitaanko tietovarastoa tapahtumankäsittelyyn. Relaatiotietokantojen hallintajärjestelmät sisältävät yleensä tapahtumien hallintaominaisuuksia. Jollei tietovarastoratkaisu sisällä tapahtumanhallinnan tukea ja tiedon käsittelytarve edellyttää sitä, tapahtumanhallinta täytyy toteuttaa sovelluslogiikan tasolla. Tietovarastoratkaisun valintakriteerit johtuvat kuitenkin pitkälti tietovaraston ulkopuolisista seikoista. Tietohallintostrategia, teknologian kypsyys, osaamisen saatavuus, markkinatilanne, liiketoimintatarpeet ja muu kokonaisarkkitehtuuri sekä käytettävät tiedonsiirtoformaatit ovat asioita jotka vaikuttavat tietovarastoratkaisun valintaan. Organisaation keskeisimpänä tehtävänä on näiden elementtien lähtökohtien, painopisteiden ja kehittämistarpeiden tunnistaminen. Organisaatiossa täytyy tunnistaa myös se osaaminen jota tarvitaan näiden elementtien hahmottamiseen kulloisessakin ratkaisutilanteessa.

Asiasanat
Tietojenkäsittely, tietovarasto, tietokanta, data, big data, pilvipalvelut, hankinta, tietohallinto.

Abstract

Haaga-Helia ammattikorkeakoulu Oy

Author
Tuula Erkinheimo

Degree Programme
Master's Degree Programme in Information Systems Management

The title of the thesis
The criteria for selection of data store solutions

Number of pages
47

The study focuses on criteria to be taken into account when selecting a data store solution. The criteria are intended to be a practical tool for situations where a data store solution is to be selected. Theoretical justification for the criteria is discussed. A closer examination is focused on the new data store solutions that have been created in order to approach Big Data.

Data processing needs vary according to the availability of the data, identification / non-identification, as well as the need for read- / write -operations. The Big Data phenomenon has brought along a new approach to data storage which eventually resulted in a number of new tools for processing large, varied and complex masses of data. Different types of data store solutions have an array of features that correspond to a variety of data processing needs.

The study shows that data models have evolved over time. Nowadays the most widely used is the relational data model. Data model types include a hierarchical data model, network data model, relational data model, object model, and data models which are used in NoSQL data stores. Various NoSQL –data store types have been developed in the 2000s in order to respond to the needs of Big Data –phenomenon and cloud services. The term ”NoSQL” refers to data stores for which SQL is not the only option in manipulating languages. Typical properties of NoSQL data stores are scalability of information horizontally across multiple servers, a tradeoff between transaction management and performance / scalability, as well as better support for non-structured information than that of the relational databases. NoSQL data stores are often schemaless or have a flexible schema. NoSQL data stores can be classified by their data models. One commonly used classification is the division of key-value stores, columnar stores, document stores, and graph stores.

The study concludes that the organization is to consider whether the data store is needed for transactional processes. Relational database management systems generally include transaction management features. If data processing needs require transaction management, and the data store solution does not support that, transaction management must be implemented in the application logic level. Selection criteria for data store solution are, however, largely due to aspects outside of the data store. Information management strategy, technology maturity, enterprise architecture and data transfer formats used are elements that influence the choice of a data store solution. The organization's main task is to identify the baseline, priorities and development needs of those elements. The organization must also recognize skills needed to perceive these elements in situations where data store decisions are made.

Keywords
Data processing, data store, database, data, Big Data, cloud services, aqcuirement contracts, data management.

Sisällys

Kuvat

1 Johdanto

1.1 Tavoite

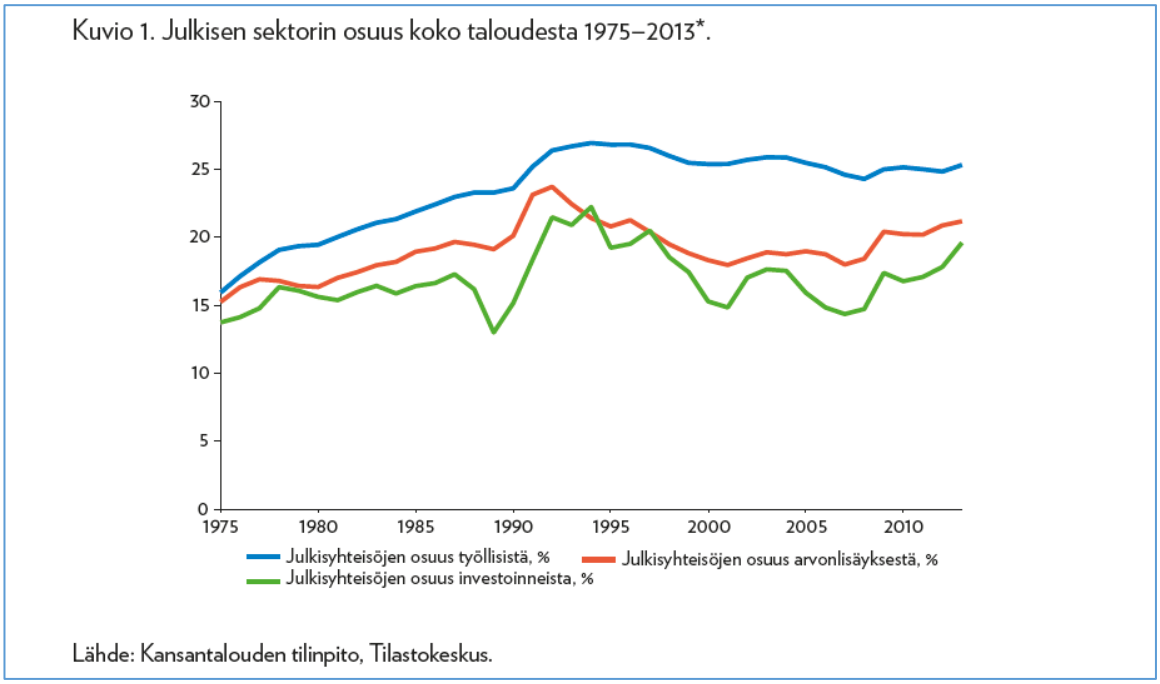
Tutkimuksen tavoitteena on tuottaa tietoa, jonka avulla on mahdollista tiivistää tietovarastoratkaisun valinnassa huomioon otettavat seikat kriteeristöksi. Tutkimuksessa tuotettava kriteeristö on tarkoitettu käytännön apuvälineeksi tietovarastoratkaisujen valintatilanteissa. Organisaation toimintatavoista riippuen kriteeristön käyttäjinä voivat olla tietohallinnon johto tai tekniset arkkitehdit. Tavoitteena on käytännön apuväline/menettely, jonka avulla voidaan esim. kilpailutustilanteissa valita oikea lähestymistapa ja tehdä perusteltu valinta erilaisten ratkaisujen välillä.

Tutkimuksella pyritään löytämään teoreettisia perusteluja tuotettavalle kriteeristölle sekä täydentämään olemassa olevaa tietoa tietovarastoratkaisuista ja erityisesti uusista vaihtoehdoista joita Big Data – lähestymistavan myötä on luotu. Tutkimuksen tutkimusongelma on ”Mitä näkökulmia on huomioitava tietovarastoratkaisun valinnassa ja mitä uutta Big Data – lähestymistapa tuo mukanaan näihin näkökulmiin?”. Tutkimusongelmaa voidaan purkaa pienempiin kysymyksiin: Millaisissa olosuhteissa uudet tietovarastoratkaisut ovat parempia kuin relaatiotietokannat? Millaisissa olosuhteissa uusia ratkaisuja kannattaa ryhtyä tutkimaan tarkemmin?

1.2 Tausta

Julkishallinnossa, kuten hallintotoiminnassa yleisemminkin, käsitellään suuria tietomassoja. Suuri osa julkishallinnon henkilöstöstä toimii käytännön työtehtävissä, esim. hoito- tai suojelutyössä. Tietyissä julkishallinnon tehtävissä korostuu kuitenkin tiedon käsittely. Yhteiskunnan alfabetisoituminen ja digitalisoituminen näkyy myös julkishallinnon työssä siten, että sellaisen tehtävän suorittaminen joka aiemmin edellytti konkreettisia toimenpiteitä, voidaan nykyään hoitaa tiedon käsittelynä. Siirtyminen konkreettisista toimenpiteistä tiedon käsittelyyn johtuu myös hallinnon tehostamistarpeesta. Tehostamista tapahtuu esim. siten, että työvoimavaltainen toiminta muutetaan tietopainotteiseksi työksi. Tällaisesta kehityksestä voidaan mainita esimerkkeinä oikeusprosessin muuttaminen suullisesta käsittelystä kirjalliseksi käsittelyksi tai lupakäsittelyn/tarkastusprosessin muuttaminen paikan päällä tapahtuvasta edellytysten tutkimisesta / tarkastamisesta kirjallisiin dokumentteihin ja digitaaliseen tietoon perustuvaksi tutkimiseksi/tarkastamiseksi.

Tietoa on mahdollista käsitellä koneellisesti, mikä vähentää työvoiman tarvetta. Julkishallinnon tehtävät ovat lisääntyneet erityisesti 1960- ja 70-luvuilla. Silloin luotiin nopeassa tahdissa mm. työeläkejärjestelmä, peruskoulu, laaja sosiaaliturva, sairausvakuutus ja lopuksi päivähoitojärjestelmä (Temmes, 2006.). Samalla näiden palveluiden hoitamista varten oli luotava hallinnolliset puitteet – rakenteet, säännöt ja henkilöstö. Olli Savelan mukaan julkisen talouden kokoa ja osuutta kansantaloudessa voi kuvata useilla mittareilla (Savela 2015, 14.). Savela on kuvannut julkisen sektorin osuutta koko taloudesta 1975–2013 kolmella mittarilla mitattuna kuviossa 1 (Savela 2015, 15.):



Kuva 1: Julkisen sektorin osuus koko taloudesta 1975-2013 (Savela 2015, 15)

Verohallinto on eräs Suomen julkishallinnon suurimmista tietojenkäsittelyorganisaatioista. Verohallinto on myös tehostanut toimintaansa merkittävästi koko 2000 – luvun ajan. Tämä tehostaminen ilmenee mm. henkilöstömäärän vähenemisenä. Verohallinnon vuoden 2014 henkilöstömäärän toteutuma oli 4.847 henkilötyövuotta (Verohallinnon tilinpäätös 2014), kun henkilötyövuosien lukumäärä vielä vuonna 2004 oli 6 266 (Verohallinnon tilinpäätös 2006). Eräänä mahdollistajana tälle kehitykselle on ollut tiedon käsittelyn tehostuminen. Verohallinnossa on tehty työtä, joka jatkuu edelleen: siirtyminen digitaaliseen tiedon käsittelyyn manuaalikäsittelyn sijasta, tiedonkäsittelyjärjestelmien kehittäminen, uudet tiedon lähteet, uudet välineet ja menettelyt tiedon välittämiseen ja varastoimiseen, uudet tavat käsitellä tietosisältöjä. Kaiken tämän taustalla on Verohallinnon toiminta-ajatus, joka on verotuksen toteuttaminen oikean määräisenä ja oikeaan aikaan yhteiskunnan toimintojen rahoittamiseksi (Verohallinnon strategia 2013–2018). Verotuksen toimittamiseksi Verohallinto tarvitsee käyttöönsä paljon tietoa. Tietoa tarvitaan mm. verovelvollisilta itseltään erilaisten ilmoitusten muodossa. Verohallinto on kuitenkin jo 1900-luvulta saakka pyrkinyt

vähentämään verovelvollisten hallinnollista taakkaa siten, että se kerää mahdollisimman suuren osan henkilöiden verottamiseen tarvittavasta tiedosta muilta tahoilta kuin henkilöiltä itseltään. Hallinnollisen taakan vähentämiseen tähtää myös julkisten palveluiden digitalisointi, joka on eräs nykyisen hallituksen kärkihankkeista. Hankkeen tavoitteena on mm. se, että julkinen hallinto sitoutuu kysymään samaa tietoa kansalaisilta ja yrityksiltä vain kerran (Sipilän hallituksen ohjelma, 24). Verotusta varten tietoja kerätään esim. palkkojen, eläkkeiden ja korkojen maksajilta (Verohallinnon päätös yleisestä tiedonantovelvollisuudesta). Nykyään yksityishenkilöistä suurin osa on veroehdotusmenettelyssä. Veroehdotusmenettelyssä Verohallinto lähettää verovelvolliselle muilta tahoilta kerättyjen tietojen pohjalta tehdyn verotusehdotuksen, johon verovelvollisen tarvitsee reagoida vain siinä tapauksessa, että ehdotukseen merkityt tiedot eivät pidä paikkaansa. Verovelvollisen ei siis tarvitse itse ilmoittaa tietoja lainkaan veroehdotuksen pohjaksi. Veroehdotusmenettelyn on mahdollistanut laaja tietojen kerääminen muilta tahoilta. Verotuksen toimittamisen ohessa Verohallinto myös tuottaa paljon tietoa. Yksittäiselle kansalaiselle tai yritykselle tuotetaan luonnollisesti tietoa hänen/sen omasta verotuslaskennasta verotuspäätösten ja -laskelmien muodossa, sekä ohjeistusta ja neuvontaa verotukseen liittyvistä asioista. Veronsaajille eli mm. valtiolle, kunnille ja seurakunnille, tuotetaan tietoa näiden saamista verotuloista, sekä ennusteita verotulojen jatkokehityksestä. Verohallinto tuottaa siten paljon sellaista tietoa, jota tarvitaan yhteiskunnallisen päätöksenteon pohjaksi.

Verohallinnossa on tiedon käsittelylle erilaisia tarpeita, jota tietovarastoratkaisuiden täytyy tukea. Erilaisia käyttötarpeita ovat esim. verotuspäätösten tekeminen, vertailutietojen käsittely, tutkimus, asianhallinta ja dokumenttienhallinta. Uudet tiedon käsittelyn lähestymistavat sekä teknologian kehittyminen luovat uusia valintamahdollisuuksia ja ratkaisuiden jatkuvat uusimistarpeet luovat jatkuvaa tarvetta eri vaihtoehtojen vertailulle. Verohallinnossa onkin tunnistettu tarve saada käyttöön välineitä joiden avulla ratkaisun valinta helpottuu ja nopeutuu nykyisestä.

2 Tutkimusmenetelmät

Tutkimuksellisia lähestymistapoja voidaan tyypitellä erilaisilla luokituksilla. Kuten Hirsjärvi, Remes ja Sajavaara painottavat (Hirsjärvi, Remes & Sajavaara 2009, 128), tutkimusstrategian samoin kuin yksittäisten tutkimusmetodienkin valinta riippuu valitusta tutkimustehtävästä tai tutkimuksen ongelmista. Tutkija tekee valintoja koko tutkimuksen ajan, ja nämä valinnat vaikuttavat tutkimuksen lopputulokseen. Tutkija tekee kuitenkin valintoja myös ennen tutkimuksen aloittamista. Jotkin valinnoista ovat tietoisia ja osan valinnoista hän saattaa tehdä tiedostamatta. Hirsjärvi, Remes ja Sajavaara ovat luetelleet tutkijan ennen aineiston keruuta tekemiä valintoja seuraavasti (Hirsjärvi ym., 120):

Ongelmanasettelu

- Kuinka täsmällisesti voin nimetä ongelman?
- Miten jäsennän ongelman?
- Miten muotoilen ongelman selvästi ja ymmärrettävästi?

Tieteenfilosofiset valinnat

- Miten ymmärrän tutkittavan kohteen? (Ontologinen kysymys.)
- Miten ajattelen saavani tietoa? (Epistemologinen kysymys)

Menetelmävalinnat

- Mitkä menetelmät tuovat parhaiten vastauksen asettamaani ongelmaan?
- Mitkä vaihtoehdot tulevat kyseeseen?
- Miten perustelen valintani?
- Mitä aineistoa on tärkeää kerätä?

Teoreettinen ymmärtäminen

- Mitkä teoriat liittyvät tutkimuksen aiheeseen?
- Mikä suhde tutkimuksella on teoriaan? (Teorian testaaminen, teorian rakentelu, praktinen teoria)
- Mitä pidän tutkimuksen avainkäsitteinä?
- Miten määrittelen käsitteet? (Teoreettisesti operationaalisesti jne.)
- Tulevatko tässä tutkimuksessa hypoteesit kyseeseen?
- Miten asetan mahdolliset hypoteesit?

Eräs tapa tutkimusstrategioiden tyypittelyyn on jako soveltavaan tutkimukseen ja perustutkimukseen. Soveltavassa tutkimuksessa tyypillisesti ratkaistaan konkreettisia ongelmia, ja perustutkimuksessa hankitaan tietoa jolle ei välttämättä tutkimushetkellä ole tiedossa käyttötarkoitusta. Tämä tutkimus toteutetaan tietyn käytännön lopputuloksen tuottamiseksi, joten kyseessä on soveltava tutkimus.

Tutkimusmenetelmät jaetaan usein karkealla tasolla kahteen tyyppiin: määrällisiin l. kvantitatiivisiin ja laadullisiin l. kvalitatiivisiin tutkimusmenetelmiin. Tällä tutkimusmenetelmien karkealla jaottelulla pyritään kuvaamaan eri menettelyjen tyypillisiä piirteitä. Tyypittely ei siten ole poissulkeva siten, että tietty menettely kuuluisi itsestään selvästi vain yhteen kategoriaan. Kvantitatiivista ja kvalitatiivista tutkimusta ei käytännössä voi tarkkarajaisesti erottaa toisistaan. Yksittäisessä tutkimuksessa käytettävässä menetelmässä saattaa olla piirteitä määrällisestä tutkimuksesta, vaikka se pääsääntöisesti luokiteltaisiinkin laadulliseksi tutkimukseksi. Yksittäisessä tutkimuksessa voidaan myös käyttää rinnakkain kvantitatiivisia ja kvalitatiivisia menetelmiä, tai esim. kvantitatiivinen vaihe voi edeltää kvalitatiivista vaihetta. Laadullinen aineiston käsittely kätkee sisälleen kvantitatiivisen tutkimuksen menetelmiä, sillä laadullinen aineisto voidaan esim. kategorisoida, mikä on määrällisen tutkimuksen analysointimenetelmä (Kananen 2009, 29). Nykyään monet tutkijat haluaisivatkin poistaa tämän tapaisen vastakkainasettelun (Hirsjärvi ym., 131).

Hirsjärven, Remeksen ja Sajavaaran mukaan kvantitatiivisessa tutkimuksessa ovat keskeisiä muun muassa:

- teoriat
- hypoteesien esittäminen
- käsitteiden määrittely
- koejärjestelyn ja aineiston keruun suunnitelmat siten, että havaintoaineisto soveltuu määrälliseen, numeeriseen mittaamiseen
- perusjoukon määrittely ja otos
- muuttujien muodostaminen taulukkomuotoon
- aineiston saattaminen tilastollisesti käsiteltävään muotoon sekä päätelmien teko havaintoaineiston tilastolliseen analysointiin perustuen. (Hirsjärvi ym., 136)

Saman lähteen mukaan kvalitatiivisen tutkimuksen tyypillisiä piirteitä ovat mm.:

- luonteeltaan kokonaisvaltaista tiedon hankintaa
- ihmisen suosiminen tiedon keruun instrumenttina; tutkijan omat havainnot sekä keskustelut tutkittavien kanssa suuremmassa roolissa kuin mittausvälineillä hankittava tieto
- induktiivisen analyysin käyttäminen; lähtökohtana ei ole teoria tai hypoteesien testaaminen vaan aineiston monitahoinen ja yksityiskohtainen tarkastelu
- laadullisten metodien käyttö aineiston hankinnassa
- kohdejoukon tarkoituksenmukainen valinta; ei satunnaisotoksen menetelmää käyttäen
- tutkimussuunnitelman muotoutuminen tutkimuksen edetessä

- tapausten käsittely ainutlaatuisina ja aineiston sen mukainen tulkinta. (Hirsjärvi ym., 160)

Tämän tutkimuksen tarkoituksena ei ole teorian luominen tai hypoteesien testaaminen, vaan tutkimustyö tähtää käytännön työvälineen luomiseen tiettyyn ympäristöön tietylle organisaatiolle. Työvälineen luominen edellyttää aihepiiriin liittyvän tiedon kokonaisvaltaista käsittelyä ja monitahoista tarkastelua. Tutkimusta ei siten voi tehdä tilastollista menetelmää käyttäen. Tutkimus on luonteeltaan kvalitatiivinen tutkimus. Tutkimusaineisto kerätään kirjallista materiaalia hyödyntämällä, havainnoimalla ja/tai haastattelemalla. Aineiston käsittelyssä korostuvat mm. seuraavat seikat, joita Hirsjärvi mainitsee esimerkkeinä kvalitatiivisten tutkimustyyppien ryhmittelyssä (Hirsjärvi ym., 162): sisällönanalyysi, mallien löytäminen ja tulkinta.

Koska tutkimuksen tavoitteena on reaalimaailman ongelman ratkaisu, tutkimusote on konstruktiivinen. Tutkimuksen otetta kuvaa hyvin Haaga-Helian Ammattikorkeakoulun opinnäytetyön sisältö ja menetelmät – ohjeen kuvaus: Konstruktiota ei löydetä, vaan se keksitään ja kehitetään. Näkökulmana on soveltava tutkimus ja toiminnan kehittäminen. (Haaga-Helia 2010, 19)

3 Tiedon käsittelyn tarpeita

Tässä kappaleessa on esitetty erilaisia näkökulmia, joiden avulla tiedon käsittelytarpeita on mahdollista luokitella.

3.1 Tiedon saannin ajallinen tarve

Eräs tapa luokitella tiedon käsittelytarpeita on käsittelyn ajallisen vastineen saantitarve. Vastineen saantitarpeen ajallista ulottuvuutta voidaan kuvata reaaliaikaiseksi tai ajastetuksi. Verohallinnolla on tarve käsitellä tietoa sekä reaaliaikaisesti että ajastetusti. Reaaliaikaisimmillaan tiedon käsittelyn tarve ilmenee sähköisissä asiointipalveluissa sekä vuorovaikutteisissa verotussovelluksissa, joita verovirkailijat käyttävät. Näissä tilanteissa tietojärjestelmäpalvelua tai sovellusta käyttää henkilö, joka odottaa järjestelmän antamaa vastinetta toimenpiteeseensä välittömästi. Tällaisissa käyttötilanteissa jo muutaman sekunnin viive aiheuttaa käyttäjälle kokemuksen huonosta käytettävyydestä. Koistinahon mukaan saatavuus on olennaisin laatutekijä kriittisissä järjestelmissä, ja kriittisen järjestelmän saatavuus on erityisen tärkeää, jos käyttäjä tai toinen järjestelmä on riippuvainen kyseisestä järjestelmästä (Koistinaho 2013, 4).

Ajastettuun vastineen saantitarpeeseen vastataan esim. eräajojen ja tilastollisen tietojenkäsittelyn avulla. Näissä tilanteissa tietoja ei yleensä käsittele ihminen, vaan toinen järjestelmä/sovellus. Suuri osa Verohallinnon tietojen käsittelytarpeesta on ajastettua tarvetta. Esim. tuloverotuksessa verotuslaskenta tapahtuu yhden verovelvollisen osalta yleensä yhden kerran vuodessa, jolloin tieto haetaan käsittelyä varten tietovarastosta vain kerran vuodessa. Tietojen tallentaminen tietovarastoon, eli lähtötietojen kerääminen, järjestäminen ja tarkistaminen voidaan tehdä koko kalenterivuoden aikana ennen verotuslaskentaa. Tietojen keräämistä varten tietojen toimittajille on asetettu määräpäiviä joihin mennessä tiedot tulee toimittaa. Tiedon tarkistamisen prosessit voidaan järjestää kustannustehokkaasti siten, että k.o. määräpäivän jälkeen samanlaiset tarkistukset tehdään kaikkien verovelvollisten tiedoille samalla kertaa eräajojen avulla.

3.2 Yksilöidyt tiedot ja yksilöimättömät aineistot

Tiedon käsittelyn tarpeet ovat erilaisia riippuen siitä, onko kyseessä yksilöityjen tietojen käsittely vai aineistojen käsittely. Yksilöityjen tietojen käsittelyä Verohallinnossa on esim. yhden verovelvollisen yhden verovuoden tuloverotus. Aineistojen käsittelyä Verohallinnossa on esim. yhden kuukauden verokertymän laskeminen yhdelle kunta-veronsaajalle.

Yksilöidyssä tietojen käsittelyssä käsitellään siis yhden yksilön verotustietoja ja aineistojen käsittelyssä käsitellään kaikkien yksilöiden yhteen laskettuja verotustietoja. Yksilöidyssä tietojen käsittelyssä yksilöt on tunnistettava, jotta tiedot voidaan liittää oikeaan yksilöön. Aineistojen käsittelyssä ei tarvitse tunnistaa yksittäisiä yksilöitä enää sen jälkeen, kun lähtödata on muodostettu.

Yksilöityjä tietoja ja aineistoja käsitellään usein eri järjestelmillä. Järjestelmäarkkitehtuurille on tyypillistä, että yksilöityjä tietoja käsitellään useilla operatiivisilla järjestelmillä, joilla on kullakin omat tehtävänsä. Aineistojen käsittelyä varten on usein erilliset järjestelmät, joilla esim. tuotetaan raportteja tai tehdään eräajoilla päivityksiä aineistoihin, jotka ladataan yhdellä kertaa operatiivisen järjestelmän tietovarastoon.

3.3 Kirjoitusta vai lukua

Tietovarastojärjestelmiä käyttävät sovellukset voidaan jaotella esimerkiksi niiden suorittamien operaatioiden tyypin ja määrän mukaan. Tietovarastoon on tallennettava tietoja ja sieltä täytyy pystyä hakemaan tietoja. Nämä toimenpiteet täytyy siis pystyä tekemään kaikissa tietovarastoratkaisuissa. Tietovarastoratkaisun valinnan ja tietovaraston rakenteen suunnittelun kannalta on kuitenkin hyvä tunnistaa kohdistuuko tietovarastoon enemmän kirjoitus- vai lukuoperaatioita.

Liiketoiminnan operatiiviset tietojenkäsittelysovellukset, joita nykyään kutsutaan online transaction prosessing eli OLTP-järjestelmiksi, ovat taas keskittyneet kirjoitusoperaatioihin ja koostuvat enimmäkseen yksinkertaisista vain muutamien kohteiden luku- tai kirjoitusoperaatioista (Sandborg 2012, 17). Operatiiviseen käyttöön tarkoitettuun tietovarastoon täytyy usein pystyä tallentamaan yksittäisiä tietoja usein. Tällainen käyttötarve on esim. silloin kun asiakas tai organisaation työntekijä tallentaa tietoja järjestelmään käyttöliittymän kautta. Moni käyttäjä saattaa tallentaa tietoja samaan aikaan, jolloin tietovaraston hallintajärjestelmän täytyy pystyä hallitsemaan usean käyttäjän samanaikaista tallentamista.

Hakupainotteinen käyttötarve on sellaisissa tilanteissa, joissa tietovarastosta pääasiassa haetaan tietoja esim. tilastoihin, tutkimuksiin tai raportteihin. Tämän tyyppisiin tietovarastoihin tietosisältö tyypillisesti tallennetaan massatallennuksina eräajoilla tai tiedonsiirroilla. Tallennus siis ei tapahdu siten, että käyttäjä tallentaisi yksittäisen tiedon kerrallaan, vaan tallennus on koneellista. Tämän tyyppisessä tietovarastossa yksittäisen yksilön tietojen ajantasaisuudelle ja oikeellisuudelle ei yleensä aseteta yhtä suuria vaatimuksia kuin ope-

ratiivisessa tietovarastossa. Jos tämän tyyppisessä tietovarastossa yksilön tietojen ajantasaisuus ja oikeellisuus kuitenkin halutaan varmistaa, se voidaan tehdä aikatauluttamalla eräajoja ja tiedonsiirtoja. Näihin tietovarastoihin kohdistuu kuitenkin tyypillisesti suuri määrä monimutkaisia lukuoperaatioita.

Uusien Web 2.0 -sovellusten, kuten yhteisöpalvelujen sovelluksien operaatiot sijoittuvat edellä mainittujen sovellusten välimaastoon ja koostuvat enimmäkseen yksinkertaisista luku- ja kirjoitusoperaatioista (Sandborg 2012, 17).

4 Big Data

Käsitteelle Big Data ei ole vakiintunutta suomenkielistä vastinetta, joten myös suomen kielisissä esityksissä käytetään yleisesti samaa englanninkielistä käsitettä. Big Datalle ei myöskään ole olemassa vakiintunutta määritelmää. Bermanin mukaan Big Data määritellään usein kolmen V:n avulla: Volume, Variety ja Velocity (Berman 2013, Introduction xx). Näillä sanoilla kuvataan Big Datalle tyypillisiä piirteitä: suurta datamäärää, datan vaihtelevia muotoja, ja datan liikkuvuutta. Tallennetun datan määrän kasvu on arvioitu neljä kertaa nopeammaksi kuin maailmantalouden kasvu, ja tietokoneiden laskentatehon kasvu on arvioitu yhdeksän kertaa nopeammaksi (Mayer-Schönberger & Cukier 2013, 9). Big Data –käsitteistö ja -keskustelu liittyy erityisesti digitaaliseen dataan.

Digitaalisen datan määrän, nopeuden ja vaihtelun lisääntyminen ovat edellyttäneet uudenlaisia tiedon käsittelyn välineitä ja menetelmiä. Datan analysoinnissa käytetään enenevässä määrin korvikemuuttujia sekä ennustavan analytiikan välineistöä ja menetelmiä. Datan käsittelyyn on luotu uudenlaisia prosessointiteknologioita, kuten Googlen MapReduce ja sitä vastaava open-source-teknologia Hadoop. Datan varastointiin on kehitetty uuden tyyppisiä tietovarastoja, joita kuvataan tarkemmin kappaleessa 8.5.

Mayer-Schönbergerin ja Cukierin mukaan Big Data tuo tietojen käsittelyyn kolme perustavanlaatuista muutosta. Ensimmäinen muutos liittyy analysoinnissa käytettävän tiedon määrään. Aiemmin ilmiöitä on tyypillisesti analysoitu otantojen avulla. Tietyn ilmiön analysointi aloitettiin usein tiedon keruun suunnittelulla; suunniteltiin millainen otoskoko tarvitaan ja miten otanta tehdään, jotta saadaan riittävän edustava otanta ilmiön analysoinnin mahdollistamiseksi. Aiemmin otantamenettelylle ei juurikaan ole ollut vaihtoehtoja, koska dataa ei ole ollut riittävästi saatavilla ja toisaalta ei ole ollut myöskään välineitä joilla suuria datamääriä olisi pystytty käsittelemään. Niinpä datan niukkuutta on korvattu menetelmillä, ja monimutkaisia datan käsittelymenetelmiä on kehitetty jotta analyysejä voitaisiin tehdä mahdollisimman pienellä lähtödatan määrällä. Tilastotieteen eräänä päämääränä onkin mahdollisimman merkittävien löydösten tekeminen mahdollisimman vähäisestä lähtödatasta (Mayer-Schönberger & Cukier 2013, 20). Tämä onnistuu siten, että satunnaisotoksesta ekstrapoloidaan laajempi tulos. Menetelmä on hyvä, jos voidaan luottaa siihen että otos on satunnainen, mutta satunnaisuuden varmistaminen ei aina ole helppoa. Ja vaikka satunnaisotannalla saadaan hyvä kokonaiskuva koko aineistosta, sen avulla ei saada porautuvaa näkymää alaryhmiin tai erikoisryhmiin, koska näiden mukanaolon varmistaminen aiheuttaisi helposti satunnaisuuden puuttumisen. Satunnaisotanta myös suunnitellaan

siten, että siitä löytyy vastaus tiettyyn kysymykseen. Tämä tarkoittaa kääntäen myös sitä, että tuolla kyseisellä otannalla ei saada vastausta muihin kysymyksiin. Nykyään on kuitenkin mahdollista käsitellä koko käytettävissä olevaa tietomassaa eli N=kaikki. Rajoitteita ei siten ole sen enempää otosjoukon kuin tietoalkioidenkaan määrälle. Kirjoittajat muistuttavat, että satunnaisotannan menetelmä on tarkoitettu sellaisten rajoitteiden kiertämiseen, jotka estävät koko tietomassan käsittelyä, ja että satunnaisotannan konsepti on kehitetty vasta alle sata vuotta sitten (Mayer-Schönberger & Cukier 2013, 21). Sen ajan rajoitteita ei enää nykyään ole samassa mittakaavassa olemassa.

Tämä ensimmäinen muutos mahdollistaa toisen muutoksen. Kun analyysiin on käytettävissä moninkertainen tietomassa aiempaan verrattuna, tiedon täsmällisyysvaatimuksia voidaan lieventää. Voidaan siis tehdä virheiden vaihtokauppa: kun otantavirheet vähenevät, voidaan sallia enemmän mittavirheitä (Mayer-Schönberger & Cukier 2013, 13). Tämä lähestymistapa ei kata kaikkia tilanteita, vaan monissa yhteyksissä tarvitaan edelleenkin eksaktia tietoa. Esimerkiksi pankkitilitapahtumien kirjaamisessa tai henkilön verotuspäätöksen tekemisessä tiedon eksaktius ja yksittäisten tapahtumatietojen oikeellisuus tulee edelleen olemaan yhtä tärkeää kuin se on ollut aiemminkin. Verohallinnon perustehtävässä, eli verotuksen toimittamisessa, on kyse yksittäisten yksilöiden talouteen vaikuttavien viranomaispäätösten tekemisestä. Tätä tehtävää varten lähtötietojen täytyy olla eksakteja. Valtavirran tietojen käsittelyn teknologiat – esim. relaatiotietokannat - ovat kuitenkin aiemmin tukeneet pääasiassa vain ja ainoastaan tämän tyyppistä datan käsittelyä, mutta Big Data tuo tullessaan enenevässä määrin mahdollisuuksia ja tarpeita myös muun tyyppiselle tiedon käsittelylle. Mikrotason tarkkuusvaatimuksien lieventäminen makrotasolla mahdollistaa näkemyksen muodostamisen datan pohjalta. Mayer-Schönberger ja Cukier kuvaavat miten data voi olla sotkuista (messy) monella eri tavalla: yksittäiset havainnot voivat olla virheellisiä, eri tyyppistä eri lähteistä olevaa dataa voidaan yhdistää suurpiirteisesti, yhdistettävä data voi olla monessa eri muodossa, tai sotkuisuus syntyy dataa yhdisteltäessä. Sotkuisestakin datasta voidaan kuitenkin löytää informaatiota. Verohallinnossakin sotkuista dataa on mahdollista käyttää moniin sellaisiin käyttötarkoituksiin, joissa datan eksaktius ei ole välttämätöntä. Tällaisia käyttötarkoituksia voi löytyä esim. sisäisten prosessien tehostamisesta tai asiakaspalvelun parantamisesta. Sotkuisuus ei kuitenkaan ole Big Datankaan luontainen ominaisuus. Sotkuisuus johtuu vain siitä, että datan keräämisen, tallentamisen ja analysoinnin välineet ovat epätäydellisiä. Jos tilastotieteilijät aiemmin tekivät vaihtokaupan suuremmista aineistoista satunnaisuuteen, nykyään voidaan tehdä vaihtokauppa tarkkuudesta suurempaan datamäärään (Mayer-Schönberger & Cukier 2013, 41). Datan sotkuisuuden hyväksyminen tarkoittaa todellisuuden monimuotoisuuden hyväksymistä. Kaiken luokitteleminen ja mallintaminen siisteihin relaatiotauluihin on suuri työmäärä, eikä siitä saatu hyöty välttämättä ole panostuksen arvoinen. Yhden totuuden

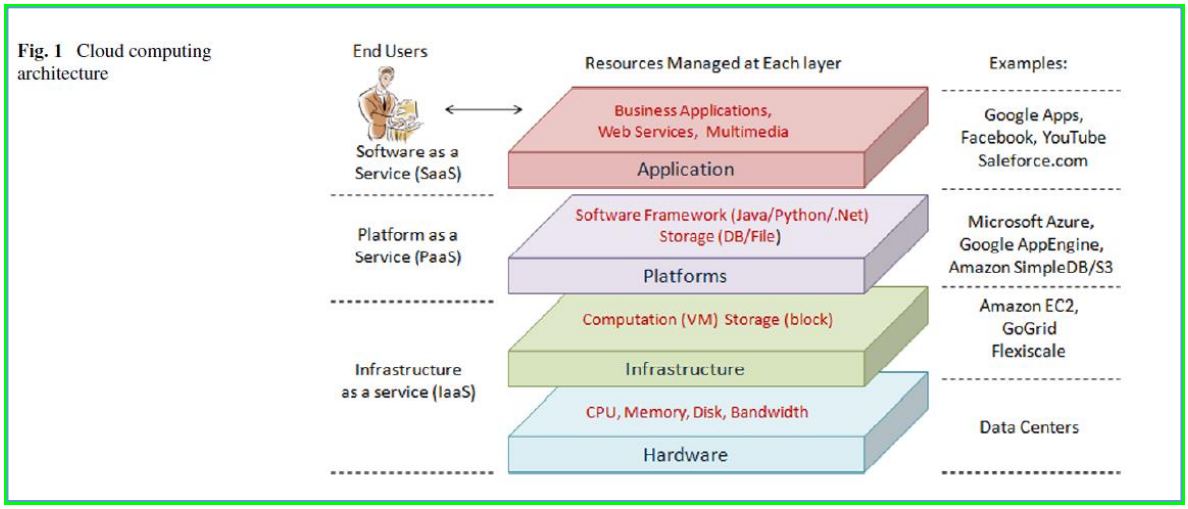
tavoittelu saattaa jopa olla haitaksi silloin kun tärkeämpi tavoite on kuitenkin ilmiön ymmärtäminen (Mayer-Schönberger & Cukier 2013, 44).

Kolmas muutos merkitsee luopumista syy-seuraussuhteen, eli kausaliteetin, etsimisestä. Kuten toinen muutos, tämäkään ei ole kaikkiin tilanteisiin sopiva lähestymistapa. Kausaliteetin etsiminen ja löytäminen tulee edelleenkin olemaan tarpeellista monenlaisissa tutkimuksellisissa konstruktioissa. Big Data on kuitenkin tuonut mukanaan uudenlaisen lähestymistavan, joka on jo osoittanut vahvuutensa. Kausaliteetin hakemisen sijasta joissain tilanteissa saattaa olla hyödyllisempää tyytyä kaavojen (pattern) ja korrelaatioiden tunnistamiseen. Big datan aikakaudella voidaan siis siirtyä hypoteesivetoisesta lähestymistavasta datavetoiseen lähestymistapaan. Tulokset saattavat olla vähemmän vääristyneitä ja tarkempia, ja ne saadaan nopeammin (Mayer-Schönberger & Cukier 2013, 55). Laskentateho on myös lisääntynyt, joten nykyään voidaan suuresta datamäärästä löytää monimutkaisempia korrelaatioita kuin aiemmin. Aiemmin on usein jouduttu tyytymään lineaaristen korrelaatioiden löytämiseen, vaikka todellisessa maailmassa ilmiöiden väliset vastaavuudet saattavat olla myös muun laisia (Mayer-Schönberger & Cukier 2013, 61). Korrelaatio on matemaattinen tosiasia, kun taas syy-seuraussuhdetta pystytään todistamaan vain harvoin. Yleensä voidaan todentaa vain syy-seuraussuhteen olemassa olo tietyllä todennäköisyydellä. Korrelaatioiden löytäminen on käytännössä nopeaa ja helppoa verrattuna esim. perinteisiin koeasetelmiin lääkekokeissa koeryhmineen ja verrokkiryhmineen (Mayer-Schönberger & Cukier 2013, 86). Aiemmin analyysi on usein tarkoittanut edeltä käsin muodostettua hypoteesia, datan keruun suunnittelua hypoteesin testaamisen mahdollistamiseksi, ja hypoteesin testaamista kerätyn datan avulla. Hypoteesi ja sen testaamiseksi tarvittava data on siis täytynyt suunnitella jo ennen datan keräämistä. Big Data –aikakaudella dataa on käytettävissä suuria määriä ja kaavojen ja korrelaatioiden löytäminen on mahdollista ilman että hypoteeseja muodostetaan etukäteen. Datan tulkintaan tarvitaan edelleenkin osaamista, koska myös löydetyt kaavat ja korrelaatiot saattavat olla satunnaisia ja siten merkityksettömiä tutkittavan ilmiön kannalta. Kuten Datavirtualization-blogissa kuvataan: ”Big Data is not always Big Information” (Van Der Lans 2015).

5 Pilvipalvelut

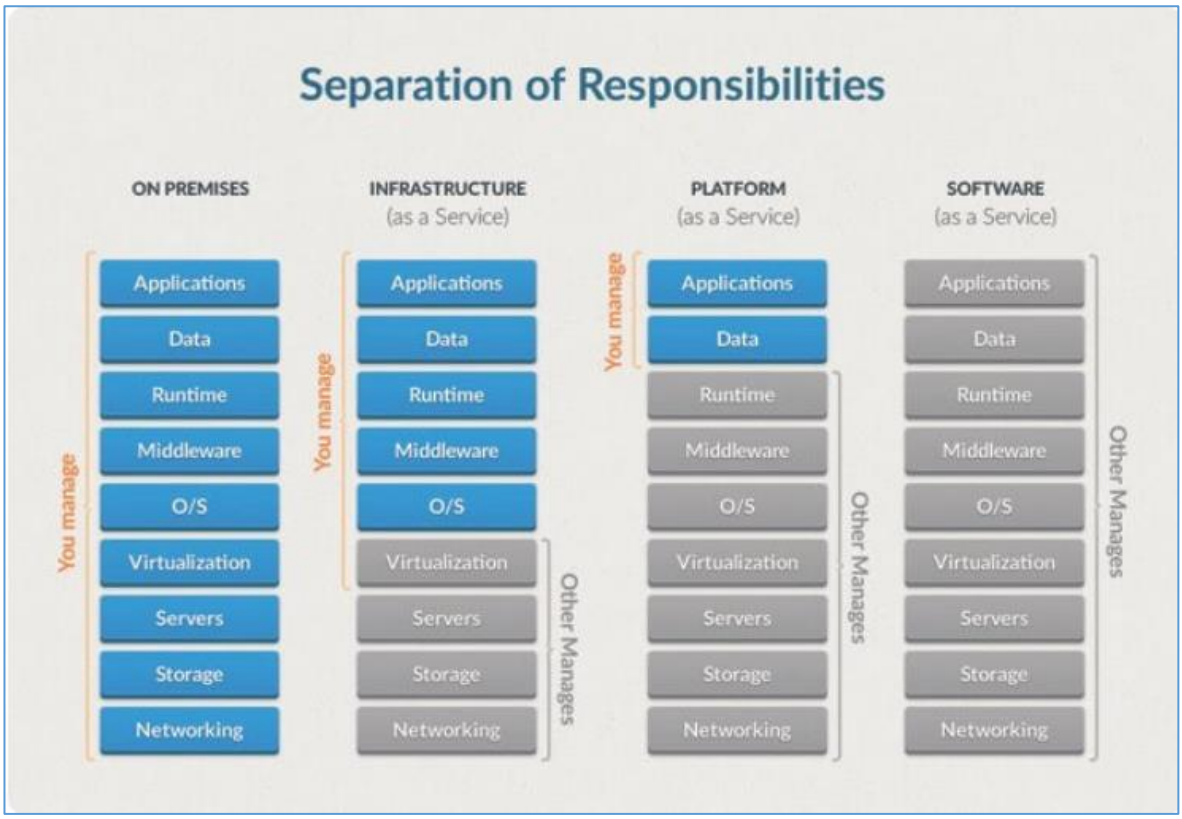
Tietoverkkojen ja tiedonsiirron ohjelmistojen kehittyminen, nopeutuminen ja yleistyminen ovat mahdollistaneet uusia palvelukonsepteja ICT-toimialalla. Erilaisia hyödykkeitä tarjotaan nykyään etäpalveluna tietoverkon kautta, eli pilvipalveluna. Pilvipalveluissa se fyysinen palvelin jolla hyödyke on, voi sijaita maantieteellisesti katsottuna missä tahansa. Hyödykkeen käyttöä tarjotaan asiakkaalle tietoverkon kautta. Asiakas ostaa pilvipalvelussa tarvitsemaansa hyödykettä vain kulloinkin tarvitsemansa määrän. Hyödyke voi olla esim. muistitilaa, laskentatehoa tai yleisten, vakioitujen ohjelmistojen käyttöä.

Pilvipalvelumallit jaetaan usein kolmeen ryhmään: IAAS, PAAS ja SAAS. Pilvipalveluiden arkkitehtuuri voidaan jakaa neljään kerrokseen Zhangin, Chengin ja Boutaban kuvauksen mukaisesti:



Kuva 2: Cloud computing architecture (Zhang, Cheng & Boutaba 2010, 9)

Christian Spoiala puolestaan kuvaa pilvipalvelumallien eroja tilaajan ja tuottajan välisen vastuunjaon näkökulmasta seuraavasti:



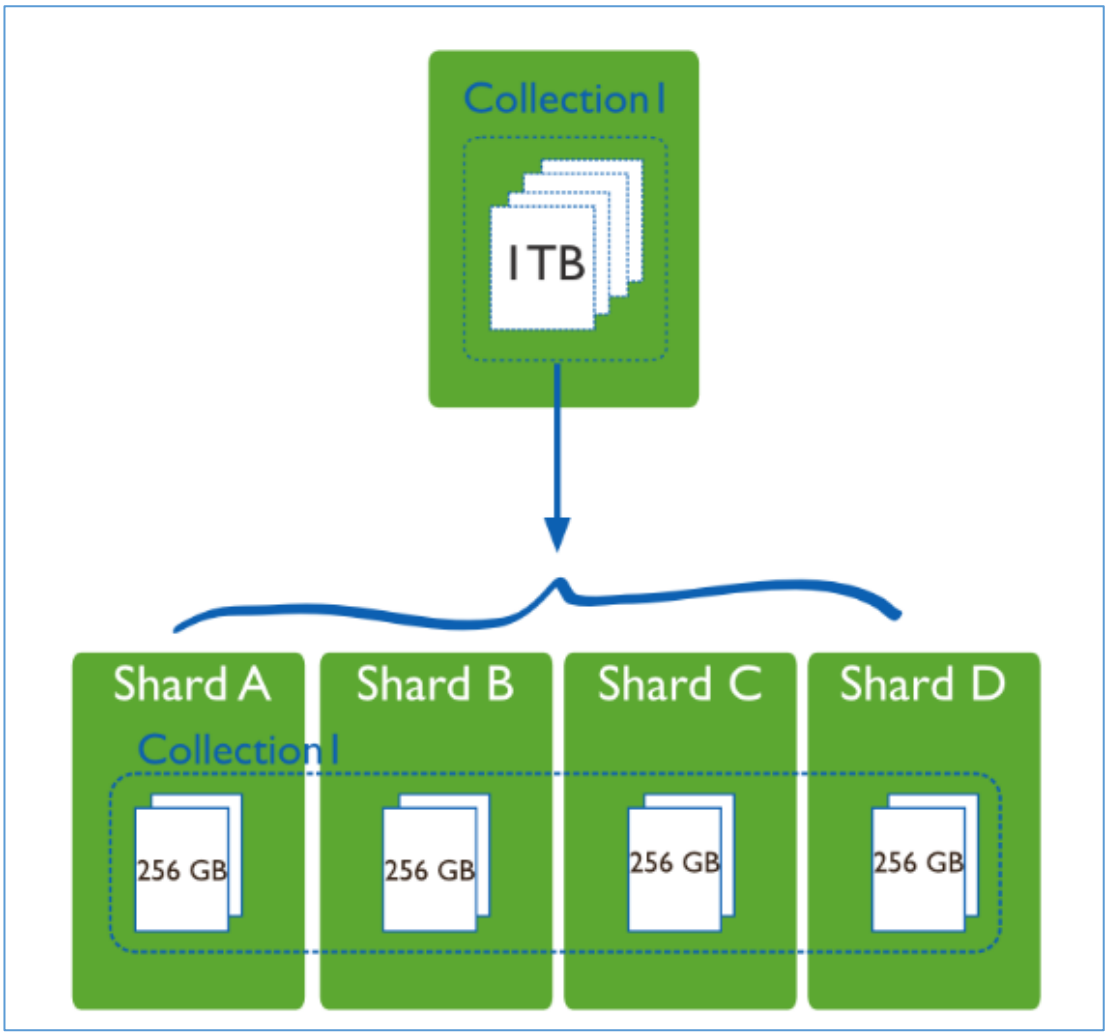
Kuva 3: Separation of Responsibilities (Spoiala 2015)

Eri pilvipalvelumallit eroavat siten toisistaan palvelun laajuuden mukaan: IAAS-mallissa palveluntuottaja tarjoaa lähinnä infrastruktuurin eli palvelimia, tallennustilaa, tietoliikennepalveluita ja käyttöjärjestelmiä. PAAS-mallissa palveluntuottaja voi tarjota myös esim. kehitysympäristöjä ja –välineitä. SAAS-mallissa palveluntuottaja tarjoaa kokonaisia loppukäyttäjien käyttämiä sovelluksia ja ohjelmistoja pilvipalveluna. Pilvipalveluna tarjottava hyödyke voi olla esim. tietovarasto, tai pilvipalvelumallilla voidaan tarjota jotain muuta hyödykettä, jonka osana on tietovarasto tai joka toimiakseen tarvitsee taustalle tietovaraston. Palveluiden tuottaminen voi myös olla kerrostunutta, jolloin kukin pilvipalveluntuottaja keskittyy omien ydinpalveluidensa tuottamiseen ja ostaa muut palvelun tuottamiseen tarvittavat osuudet toiselta pilvipalveluntuottajalta. Esim. tuottaja A tuottaa kuluttajille verkkosovelluspalveluita, joissa tehtävän laskentaosuuden tuottaja A ostaa tuottajalta B, jonka ydintuotteena puolestaan on pilvilaskenta.

Yleisiä, pilvipalveluista ostettavia vakioituja ohjelmistoja ovat esim. toimisto-ohjelmistot ja ohjelmointialustat. Vakioitujen ohjelmistotuotteiden käytön lisääntymiseen vaikuttavat organisaatioiden keskittyminen ydintoimintoihinsa ja ohjelmistotuotannon yleinen kypsyminen sekä toisaalta ohjelmistotuotanto-toimialalla tapahtuva keskittyminen. Myös pilvipalveluiden saatavuus jo itsessään lisää vakio-ohjelmistojen käyttöä – kun vakio-ohjelmisto on helposti saatavissa ja käyttöön otettavissa, ja kun kustannukset tulevat käytön mukaan,

pilvipalvelu saattaa olla asiakkaalle hyvin houkutteleva vaihtoehto. Vakio-ohjelmistojen käyttäminen tuo organisaation toimintaan luonnollisesti tiettyä jäykkyyttä. Vakio-ohjelmiston ominaisuudet mahdollistavat tietyt toimintamallit. Käyttäjän ei kannata pyrkiä muuttamaan vakio-ohjelmiston ominaisuuksia, eikä käyttäjä yleensä edes pysty niitä muuttamaan – muutoshan näkyisi kaikille k.o. ohjelmiston käyttäjille, ja voi olla että se ei olisi toivottu muutos kaikkien käyttäjien näkökulmista katsottuna. Tämä vakio-ohjelmiston käyttämisen tuottama jäykkyys saattaa edellyttää asiakkaan organisaatiossa joskus kivuliastakin oppimisprosessia, jos toimintatapana on aiemmin ollut omien, räätälöityjen ohjelmistojen, järjestelmien ja sovellusten käyttäminen. Omassa hallussa olevia, itse tehtyjä sovellutuksia on suhteellisen helppo muuttaa käyttäjien toiveiden ja muuttuvien toimintamallien mukaan. Vakio-ohjelmistojen käyttö taas edellyttää organisaation toimintamallien sovittamista vakio-ohjelmiston ominaisuuksiin.

Pilvipalvelun tarjoaja tarjoaa samaa hyödykettä useille asiakkaille. Asiakkaita voi tulla lisää tai asiakkaat voivat vähentyä. Yhden asiakkaan tarve hyödykkeen saannille voi vaihdella ajankohdittain suurestikin. Esim. laskentatehoa saatetaan tarvita suuria määriä kerran päivässä, kuukaudessa tai vuodessa, ja muina aikoina tuskin lainkaan. Jotta pilvipalvelun tarjoajat pystyvät vastaamaan tällaisiin kysynnän muutoksiin ja resurssitarpeen vaihteluihin, tarjottavien palveluiden täytyy olla hyvin skaalautuvia. Skaalautuvuus voi olla joko vertikaalista tai horisontaalista. Tuomas-Matti Soikkeli kuvaa vertikaalista ja horisontaalista skaalautuvuutta seuraavasti: ”Vertikaalisesti skaalatessa yksittäisen laitteiston tehokkuutta kasvatetaan. Päivitetään tietokone tai vaihdetaan sen komponentteja tehokkaampiin. Tässä tapauksessa käytetään usein supertietokoneita, jotka ovat erittäin kalliita. Horisontaalisesti skaalatessa hajautetaan laskentateho useiden tietokoneiden välille. Tässä tapauksessa voidaan käyttää halpoja komponentteja, jotka voidaan liittää järjestelmään vaivattomasti. Suoritustehoa on näin helppo lisätä tarpeen vaatiessa, eikä suuria kertainvestointeja tarvitse tehdä.” (Soikkeli 2015, 13.) Tietovarastoihin tallennetun datan horisontaaliseen skaalaus toteutetaan usein termillä sharding-tekniikalla (suomeksi esim. pirstaloiminen). Shardingissa tietoaineisto jaetaan (divide) useaksi aineistoksi, jotka jaetaan (distribute) useille palvelimille. Jokainen shard on erillinen itsenäinen tietokanta ja yhdessä shardit muodostavat yhden loogisen tietokannan. (Sharding Introduction) Tuotantokäytössä shard on yleensä kopio muualla sijaitsevasta tiedosta, joten tiedosta on aina olemassa kopio, jolloin mahdollisen virhetilanteen sattuessa tietokannassa oleva tieto ei vääristy (Iivonen 2014, 30].
MongoDB:n sivustolla sharding-tekniikka on esitetty kuvan 4 avulla:



Kuva 4: Sharding -tekniikka (Sharding Introduction)

Uudentyyppisiä n.s. NoSQL-tietovarastoja on kehitetty vastaamaan niihin uusiin haasteisiin, joita tietomäärän valtava kasvu, tiedon monimuotoisuus ja pilvipalvelut ovat asettaneet tietovarastoille. Abadin mukaan pilvipalveluilla on kolme tiedonhallinnan kannalta relevanttia ominaisuutta: Laskentateho on elastista, mutta vain silloin jos pystytään käyttämään rinnakkaislaskentaa. Data saattaa olla ei-luotetun tahon hallinnassa. Dataa kopioidaan usein pitkien maantieteellisten välimatkojen yli. (Abadi 2009, 2-3.) NoSQL-tietovarastoilla pystytään ratkaisemaan useita uusia haasteita, mutta nämä ratkaisut saattavat edellyttää joidenkin muiden vaatimusten lieventämistä. Lieventämisen kohteena ovat usein n.s. ACID-tapahtumanhallintaominaisuudet, jotka on kuvattu tarkemmin kappaleessa 6.1.

6 ACID, CAP ja BASE

Tietojenkäsittelytiede, kuten muutkin tieteenalat, jakautuu erilaisiin aihepiireihin. Esim. tietokantateoriassa ja hajautettujen järjestelmien teoriassa käsitellään saman tyyppisiä asioita. Tietojenkäsittelytieteessä terminologia on yleensä englanninkielistä eikä tässä tutkielmassa ole tarkoitus kehittää suomenkielisiä vastineita näille vakiintuneille englanninkielisille termeille. Joissain yhteyksissä ACID-ominaisuudet on kuitenkin suomennettu, joten niistä esitetään tässä sekä englanninkielinen että suomenkielinen termi.

Kappaleessa 6.1 on kuvattu tietokantateoriassa tunnistetut ACID-ominaisuudet ja kappaleessa 6.2 on kuvattu hajautettujen järjestelmien teoriassa tunnistetut CAP-ominaisuudet. On huomattava, että erilaisista lähtökohdista johtuen näissä ominaisuuskoosteissa viitataan samalla englanninkielisellä termillä eri asioihin. Consistency ACID-ominaisuutena ei tarkoita samaa asiaa kuin Consistency CAP-ominaisuutena.

6.1 ACID

Tietokantateoriassa tunnistettiin 1970-luvulla n.s. ACID-ominaisuudet, eli:
A (Atomicity)
C (Consistency)
I (Isolation)
D (Durability)

ACID-ominaisuuksia tarvitaan tyypillisesti tapahtumien käsittelyssä, ja relaatiotietokantojen hallintajärjestelmät tyypillisesti pitävät huolen ACID-ominaisuuksien toteutumisesta. Transaktion ACID-ominaisuudet takaavat tietokannan luotettavuuden (Silberschatz, Korth & Sudarshan 2011, luku 14).

Atomicity eli tiedon atomisuus tarkoittaa sitä, että tapahtuma ei ole onnistunut ennen kuin kaikki tapahtuman osat ovat onnistuneet. Tietokantatapahtumat, eli transaktiot, ovat usein moniosaisia. Tietokantatapahtuman esimerkkinä esitetään usein pankkitilisiirto; yksi tilisiirto koostuu vähennyksenä yhdeltä tililtä ja lisäyksenä toiselle tilille, eikä tilisiirto ole onnistunut jos tapahtuma jostain syystä keskeytyy ennen kuin molempien tilien tapahtumat on pystytty käsittelemään. Mustosen kuvauksen mukaan tämä tarkoittaa sitä, että tapahtuma

käsitellään yhtenä kokonaisuutena, eikä se siten voi onnistua tai epäonnistua osittain. Mikäli yksikin tapahtumaan kuuluvista operaatioista epäonnistuu, niin koko tapahtuma tulkitaan epäonnistuneeksi. (Mustonen 2003, 5)

Consistency eli ristiriidattomuus tarkoittaa sitä, että tieto käsitellään vain jos se on tietokannan sääntöjen mukainen. Tietokannan hallintajärjestelmä esim. tunnistaa tietynlaiset tietotyypit ja tietokantaan voidaan asettaa erilaisia rajoitteita tiedolle. Mustosen mukaan tämä vaatimus määrää sen, että tapahtuman tulee säilyttää muokkaamansa resurssin yhtenäisyys. Eli jos muokattu resurssi, esimerkiksi tietokanta, on yhtenäisessä tilassa ennen tapahtuman suoritusta, niin sen tulee olla sitä myös tapahtuman suorituksen loputtua. Tapahtuman suorittaminen ei saa aiheuttaa vahinkoa muokattavan resurssin yhtenäisyydelle. (Mustonen 2003, 5)

Isolation eli eristyneisyys tarkoittaa sitä, että tapahtumat ovat eristettyjä toisistaan; eri laskennat eivät voi muuttaa samaan aikaan samaa tietoalkiota, eikä tapahtuman käsittelyn välivaiheita näytetä muiden tapahtumien käsittelijöille. Eristyneisyys toteutetaan usein lukitusmekanismien avulla.

Durability eli pysyvyys tarkoittaa sitä, että onnistuneen tapahtumankäsittelyn jälkeen tieto tallennetaan pysyvästi eivätkä esim. mahdolliset myöhemmin tapahtuvat järjestelmävirheet vaikuta tiedon olemassaoloon tai sisältöön.

6.2 CAP

Hajautettujen järjestelmien teoriassa tunnistettiin 2000-luvulla n.s. CAP-ominaisuudet, eli:
C (Consistency)
• A (Availability)
• P (Partition-tolerance)

Hajautetuilla järjestelmillä tarkoitetaan tässä yhden tietovaraston hajauttamista usealle palvelimelle. CAP-teoreema tarkoittaa sitä, että hajautetuissa tietovarastoissa ei voida toteuttaa kaikkia kolmea vaatimusta samaan aikaan. Hajautetuissa tietovarastoissa on siis valittava mitkä kaksi vaatimusta toteutetaan. CAP-teoreeman kolme ominaisuutta tarkoittavat seuraavia asioita:

Consistency tarkoittaa sitä, että kaikilla palvelimilla on sama data samaan aikaan siten, että datan käyttäjä saa saman vastauksen riippumatta siitä miltä palvelimelta vastaus annetaan.

Availability tarkoittaa sitä, että järjestelmä vastaa aina tietopyyntöön. Vastaus ei välttämättä ole viimeisin tieto eikä sama tieto kuin muilla palvelimilla, ja vastaus voi olla vaikkapa virheilmoitus, mutta järjestelmä vastaa aina.

Partition Tolerance tarkoittaa sitä, että järjestelmän toiminta ei keskeydy vaikka yksittäinen palvelin ei olisikaan toiminnassa tai saavutettavissa.

Kun hajautetuissa järjestelmissä vain kaksi näistä kolmesta ominaisuudesta voidaan toteuttaa samaan aikaan, hajautetuissa NoSQL-systeemeissä tyypillisesti luovutaan consistency-ominaisuudesta.

6.3 BASE

CAP-teoreemaa on usein käytetty perusteena tarpeelle transaktioiden ACID-ominaisuuksia heikentävään luotettavuusmalliin (Pritchett 2008, 50). CAP-teoreeman mukaan hajautettu järjestelmä ei siis voi taata kaikkia ACID-ominaisuuksia yhtä aikaa, ja ACID-ominaisuuksien mukainen takuu voi heikentää hajautetun järjestelmän saatavuutta. Myös tietoalkioiden lukitseminen voi vaikuttaa hajautetun tietokannan saatavuuteen.

NoSQL-systeemeissä noudatetaan tyypillisesti n.s. BASE-oikeellisuusmallia, jossa:
- BA = Basically Available
- S = Soft State
- E = Eventually Consistent

Tämä tarkoittaa, että tietokannan ei tarvitse olla oikeellisessa tilassa jokaisen transaktion jälkeen, vaan riittää että tietokannan oikeellisuus saavutetaan jossain myöhemmässä vaiheessa. Kyseinen menettely on ACID-ominaisuuksien vastainen. BASE-oikeellisuusmalli siis sallii tietokannan tietojen olevan jossain vaiheessa vanhentuneita. (Sandborg 2012, 15)

7 Tietomalli

Tietomallilla voidaan tarkoittaa tiedon rakenteen ja tiedolle suoritettavan käsittelyn määrittelevää käsitteistöä (Laine 2005, 1), tai notaatiota, jolla tietoa kuvataan (Garcia-Molina, s. 17). Laineen mukaan tietoa voidaan tarkastella eri abstraktiotasoilla; käsitteellisellä, rakenteellisella ja fyysisellä tasolla. Garcia-Molinan mukaan tiedon kuvaaminen yleensä koostuu kolmesta osasta: tiedon rakenteesta, operaatioista joita tietoon voidaan kohdistaa, ja tietoon kohdistuvista rajoituksista. Jo näiden lyhyiden kuvausten perusteella voidaan havaita, että Laine ja Garcia-Molina käsittelevät termiä ”tietomalli” eri tavoilla. Tietomalli onkin yleiskäyttöinen termi, jota käytetään myös esim. rakennusalalla. Liikennevirasto tutkimuksessa Tietomallien yhteensovittaminen siltahankkeessa tietomallilla tarkoitetaan tuotteen koko elinkaaren aikaisten tietojen kokonaisuutta digitaalisessa muodossa (Järveläinen 2014, 15). Tässä tutkielmassa termillä tietomalli viitataan Laineen esittämään määritelmään ja sen mukaisiin jaotteluihin.

Laine jakaa tiedon tarkastelun abstraktiotasot käsitteelliseen, rakenteelliseen ja fyysiseen. Tällä jaottelulla hän esittelee tarjolla olevat mallit seuraavasti (Laine 2005, 7):
Käsitetaso: Entity-Relationship mallit (ER), oliomallit (UML) ja semanttiset tietomallit
Rakennetaso: Relaatiomalli, oliomallit, XML, hierarkkinen malli ja verkkomalli
Fyysinen taso: Toimittajakohtaiset käsitteet
Koska tämän tutkielman käytännön lopputuotteena on valintakriteeristö käytännön valintatilanteita varten, tässä kuvataan sellaisia malleja, joilla on merkitystä käytännön valintatilanteissa. Tämä tarkoittaa sitä, että tässä tutkielmassa keskitytään rakennetason malleihin. Käsitetason mallilla ei ole vaikutusta tietovaraston valintaan, koska esim. sama käsitetason ER-malli voidaan usein käytännössä toteuttaa millä tahansa rakennetason mallilla. Vainio kuvaa eri tasoihin liittyviä kaavioita näin: ”Fyysiseen tasoon liittyy fyysinen kaavio, joka kertoo, miten tieto on organisoitu varsinaiseen tallennusmediaan, esimerkkinä tietuejärjestys. Käsitteelliseen tasoon liittyy käsitteellinen kaavio, joka on abstraktio kohdealueesta tietomallin kuvaamana. Vaikka fyysisessä kaaviossa kohdealueen tiedot voivat olla tallennettuna useampaan mediaan tai eri muodoissa, käsitteellinen kaavio kokoaa nämä tiedot yhdeksi kokonaisuudeksi esittäen tietokannan loogisen sisällön. Ulkoiseen tasoon liitetään näkymät, joiden kautta päästään määrättyihin käsitteellisen tietokannan osiin. […] Usein tietokantaa suunniteltaessa mallintaja käyttää aluksi käsitteellistä tietomallia ja muuttaa tämän käsitteellisen tietomallin loogiseksi tietomalliksi. Käsitteellisiä tietomalleja ovat muun muassa ER-malli (Entity Relationship) ja SDM (Semantic Data Mo-

del). Loogisia tietomalleja ovat muun muassa relaatio-, verkko-, hierarkkinen, olio- ja deduktiivinen malli. [viitteet]” (Vainio 2012, 6)

Tiedon mallintaminen tarkoittaa tiedon kuvaamista, ja tätä mallintamista voidaan tehdä erilaisia käyttötarkoituksia varten. Tietotekniikassa tiedon mallintamisen tavoitteena on tietovaraston rakentaminen tai hankkiminen, ja tietovarastossa olevan tiedon hyödyntäminen. Tietoa siis mallinnetaan, jotta se on mahdollista tallentaa tietovarastoon, jotta sitä on mahdollista hakea tietovarastosta, ja jotta sen käsitteleminen pystytään automatisoimaan. Tiedon mallintaminen tarkoittaa tiedon kuvaamista. Tiedon mallintaminen perustuu sen ja sen kuvaaman todellisuuden analyysiin, jonka tuloksena saadaan joukko entiteettejä, niiden välisiä yhteyksiä ja niiden ominaisuuksia. Tietomalli on siis malli tiedosta, ei todellisuudesta, jota se mallintaa vain epäsuorasti.

Kappaleessa 8 kuvataan tietovarastomallien kehitystä. Uudemmissa n.s. NoSQL-tietovarastoissa tiedon mallintamista ei aina käytetä tietovaraston rakentamisvaiheessa, jolloin tiedon mallintaminen voidaan siirtää ajallisesti eteenpäin. Näissä tietovarastoissa tietoa ei välttämättä mallinneta tietyn mallin mukaisesti, eikä kaikkea tietovarastossa olevaa tietoa mallinneta välttämättä lainkaan.

8 Tietovarastotyyppejä

Tietokanta (database) voidaan karkeasti määritellä jotakin käyttötarkoitusta varten laadituksi kokoelmaksi toisiinsa liittyviä säilytettäviä tietoja. Tämän määritelmän mukaan mitä tahansa säilytettävää tietokokoelmaa voidaan pitää tietokantana. (Laine 2000, 1) Toisaalta esim. Vaasan yliopiston opintomateriaalin mukaan tietokanta on tietotekniikassa käytetty termi tietovarastolle (Vaasan yliopisto, 2). Laineen mukaan tietokantaan usein liitetäänkin sen teknisiin ominaisuuksiin liittyviä lisävaateita, joiden perusteella tietokanta eroaa perinteisesti ohjelmointikielissä tarjolla olevasta tiedostosta (file) (Laine 2000, 1). Tässä tutkielmassa käytetään termiä tietovarasto, koska tässä tutkielmassa käsitellään myös sellaisia tietovarastoratkaisuja, joissa ei ole tietokannan hallintajärjestelmää. Tietokannan hallintajärjestelmällä tarkoitetaan tässä niitä toiminnallisia ominaisuuksia, jotka usein liitetään tietokanta-termiin tietotekniikan käsitemaailmassa. Tietokantatuotteisiin nämä ominaisuudet sisältyvät nykyään oletusarvoisesti. Tietokannan hallintajärjestelmä on sellainen joukko ominaisuuksia/ohjelmia, joka mahdollistavat tietokannan keskitetyn luonnin ja ylläpidon. Tämä keskitetty luonti ja ylläpito tarkoittaa mm. tietokannan rakenteen ja tietotyyppien määrittelyä, tietokannan rakentamista sekä tietokannan tiedonhaku ja –tallennusmenettelyitä.

Tässä tutkielmassa käsitellään myös sellaisia tietovarastoja, jotka eivät ole tietokantatuotteita ja joissa ei ole tietokannan hallintajärjestelmää. Viittauksissa ja sitaateissa käytetään kuitenkin termiä ”tietokanta” aina silloin kun alkuperäinen kirjoittaja on käyttänyt tätä termiä.

Tietokantojen ja tietokantatuotteiden kehittäminen on saanut alkunsa 1960-luvulla, jolloin alettiin ohjelmallisesti käsitellä niin suuria tietomääriä että ne eivät mahtuneet tietokoneen käytettävissä olevaan muistiin. Ilmeinen ratkaisu oli ylimääräisen tiedon tallentaminen ulkoiseen tietovarastoon, eli magneettilevylle tai –nauhalle. Tiedon käsittely ulkoisesta tietovarastosta on kuitenkin huomattavasti hitaampaa kuin tiedon käsittely tietokoneen muistista, joten ulkoisten tietovarastojen alkuvaiheessa teknisen kehittämisen painopiste oli saantinopeuden lisäämisessä. Eri sovellukset saattoivat käsitellä samaan ulkoiseen tietovarastoon tallennettua tietoa, mutta ongelmaksi muodostui se, että eri sovelluksilla oli jokaisella oma näkymänsä tietoon. Tietojenkäsittelyn historian tässä vaiheessa arkkitehtuuri ei ollut kerrostunutta, vaan jokainen sovellus yleensä kuvasi itse sisäisesti käyttämänsä tiedot, niiden järjestyksen ja rakenteen. Tiedon tallentamisen ulkoistaminen edellytti tietovaraston järjestyksen ja rakenteen kuvaamista siten, että kaikkien sitä käyttävien sovellusten ei tarvinnut tehdä tätä erikseen. Tämä toisaalta tarkoitti myös sitä, että kaikki ulkoista tietovarastoa käyttävät sovellukset olivat sidottuja tietovaraston vakiojärjestyk-

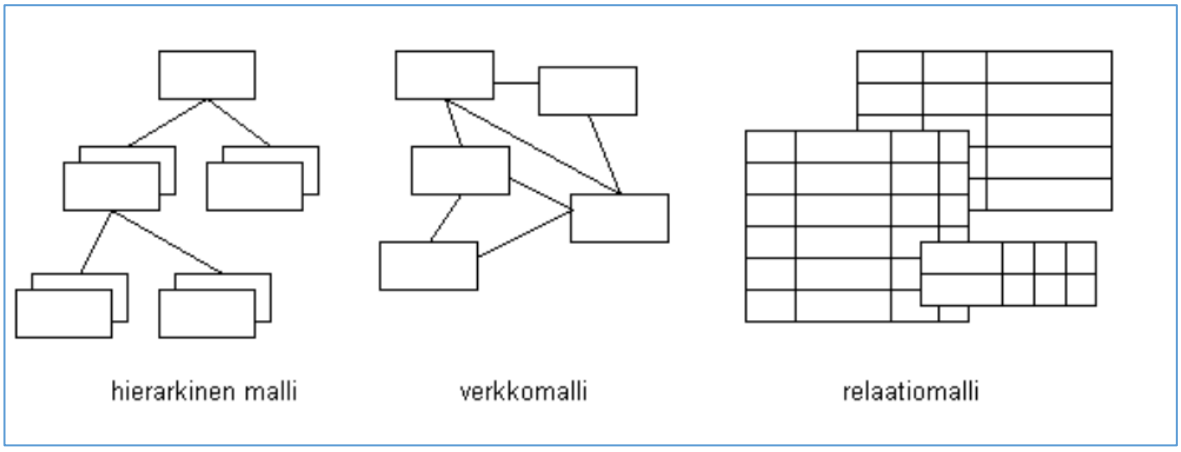
seen ja –rakenteeseen. Tällöin järjestys ja rakenne tuli kuvata yleisesti tunnetulla tavalla, jotta eri sovellukset voivat käyttää samaa tietovarastoa.

8.1 Hierarkkinen tietomalli ja verkkotietomalli

Varhaiset tietokannan hallintajärjestelmät perustuivat hierarkkiseen tietomalliin. Hierarkkisessa tietomallissa (hierarchical data model), jota kutsutaan myös NF²-tietomalliksi (non-first normal form), on rakenteen pääpiirre hierarkia (Järvelin & Niemi 1990, 4). Hierarkkisessa mallissa jokaisella tietueella on isätietue ja sillä voi olla yksi tai useampia lapsitietueita. Vainio kuvaa, että tietuetyyppien välisiä suhteita kuvataan vanhempi-lapsi –suhdetyypillä, joka on 1:n –suhde vanhemmasta lapseen. Suhteeseen liittyy aina yksi vanhempi-tietue ja mahdollisesti useita lapsi-tietueita. (Vainio 2012, 15)

Verkkotietomalli paransi hierarkkista tietomallia siten, että tietomallissa oli mahdollista esittää hierarkkisen järjestyksen lisäksi myös lapsitietueiden välisiä keskinäisiä suhteita. Verkkotietomallissa (network data model) tieto jäsentyy tietuetyyppeihin ja näiden välisiin yhteystyyppeihin, jotka muodostavat verkoston (Järvelin & Niemi 1990, 4). Ensimmäisenä verkkomallinen esitteli CODASYL-komitea (Conference on Data Languages). Vainion kuvauksen mukaan ”verkkomallin peruskäsitteitä ovat tietuekokoelma, tietue ja tietoalkion arvo sekä tietuetyyppien väliset suhteet. Verkkotietokanta koostuu tietuekokoelmista. […] Tietuetyyppien välisiä suhteita nimitetään verkkomallissa kokoelmatyypeiksi. nämä kokoelmatyypit kuvaavat 1:n-suhteet kahden tietuetyypin välillä. […] Kokoelmatyyppi ei mahdollista n:m-suhteita, vaan nämä korvataan kahdella suhteella ja täydentävällä tietuetyypillä.” (Vainio 2012, 12)

Ossi Nykänen on kuvannut seminaariesitelmässään hierarkkisen tietomallin, verkkotietomallin ja relaatiotietomallin eroja seuraavan kuvan avulla:



Kuva 5: Erilaisia tietokannan tietomalleja (Nykänen 1996)

8.2 Relaatiotietovarastot

Hierarkkiset tietokannat ja verkkotietokannat tarjosivat hyviä ratkaisuja oman aikakautensa tiedonhallinnan ongelmiin, ja mallien yksinkertaisuus tuotti tietojärjestelmiin myös vakautta ja luotettavuutta. Hierarkkisissa tietokannoissa ja verkkotietokannoissa oli kuitenkin vielä paljon puutteita. Esim. tiedon etsiminen vaati monimutkaisten puumaisten rakenteiden läpikäyntiä. E.F. Codd:in vuonna 1972 julkistama relaatiomalli sai aikaan suuren kehitysaskeleen tiedonhallinnassa. Relaatiotietokanta onkin nykyään käytetyin tietokantamalli, ja se on käytännössä syrjäyttänyt hierarkkisen mallin ja verkkomallin. Relaatiotietokantaa voidaankin hyvällä syyllä kutsua yleiskäyttöiseksi tietovarastotyypiksi. Sandborgin mukaan relaatiotietokantajärjestelmät ovat osoittautuneet alusta lähtien toimiviksi ja luotettaviksi ratkaisuiksi esimerkiksi erilaisten perinteisten liiketoimintaan liittyvien tietojenkäsittelytoimintojen (business data processing), kuten kirjanpidon, laskutuksen ja varastohallinnan sovelluksien käytössä (Sandborg 2012, 16). Relaatiotietomallin perustana on näkemys tietokannasta joukkona tietoalkioiden muodostamia matemaattisia relaatioita.

Ollikainen on kuvannut oppimateriaalissaan relaatiomallin seuraavasti (Ollikainen):

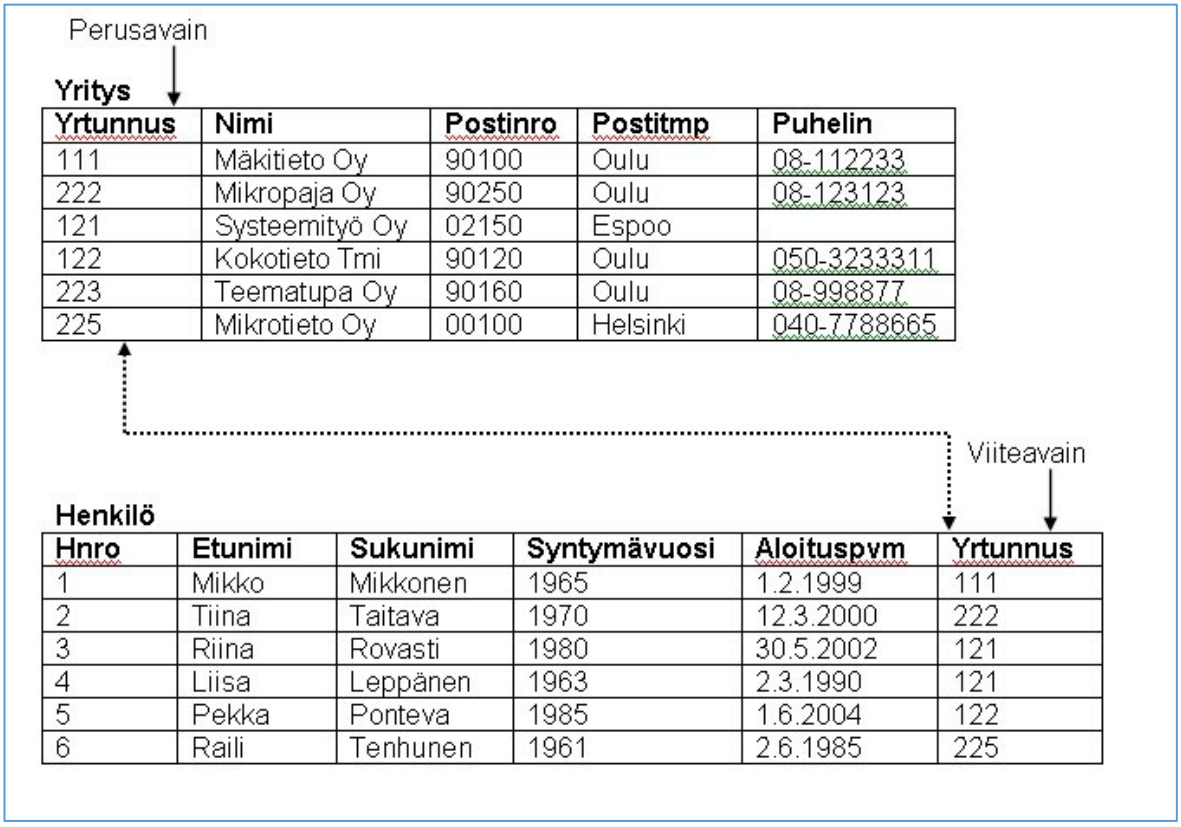
> Relaatiolla tarkoitetaan nimettyä, kaksiulotteista tietojoukkoa (taulukkoa/taulua).
>
> Relaatio täyttää seuraavat ehdot:
>
> 1. Relaatiolla (taulukolla) on nimi, joka erottaa sen muista relaatioista (taulukoista)
> 2. Relaatiolla on attribuutteja (sarakeotsikoita), joista jokaisella on relaation (taulukon) sisällä yksikäsitteinen nimi.
> 3. Relaatio (taulukko) sisältää monikoita (rivejä), joista kukin koostuu attribuuttien arvoista.
> 4. Relaation (taulukon) kaikki rivit ovat erilaisia.
> 5. Attribuuttien (sarakkeiden) ja monikoiden (rivien) järjestyksellä ei ole väliä.
> 6. Kussakin monikossa kukin attribuutin arvo (solun sisältö) on atominen eli se sisältää yhden, yksikäsitteisen arvon.
>
> Taulun pääavaimella tarkoitetaan sellaista saraketta (tai sarakejoukkoa), jonka arvot erottelevat taulun rivit toisistaan yksikäsitteisesti.
>
> Viiteavainten avulla voidaan yhdistää eri taulukoissa olevia tietoja. Viiteavain viittaa toisen taulun pääavaimeen.
>
> Sarakkeen arvoalueella tarkoitetaan niiden arvojen joukkoa, jotka sarake voi saada.
>
> Avainrajoite määrää sen, että kaikilla taulun riveillä on kelvollinen pääavaimen arvo, joka ei ole sama kuin minkään muun pääavaimen arvo.

Viiteavaimen arvon on aina esiinnyttävä kohdetaulun jonkin monikon pääavaimen arvona.

Tietokantaa suunniteltaessa tietokanta pyritään saattamaan ns. kolmanteen normaalimuotoon, jossa tietokannan taulujen rakenteella on toivottuja ominaisuuksia. Kolmanteen normaalimuotoon päästään ensimmäisen ja toisen normaalimuodon kautta.

Taulujen saattamista kolmanteen normaalimuotoon kutsutaan tietokannan normalisoinniksi. Ehtoja rikkovat taulut pilkotaan pienemmiksi tauluiksi, kunnes ehdot toteutuvat.

Relaatiotietovarasto voidaan kuvallisesti esittää tauluina ja taulujen riveinä, kuten Maarit Hiltunen esittää kuvassa 6:



Yrtunnus	Nimi	Postinro	Postitmp	Puhelin
111	Mäkitieto Oy	90100	Oulu	08-112233
222	Mikropaja Oy	90250	Oulu	08-123123
121	Systeemityö Oy	02150	Espoo	
122	Kokotieto Tmi	90120	Oulu	050-3233311
223	Teematupa Oy	90160	Oulu	08-998877
225	Mikrotieto Oy	00100	Helsinki	040-7788665

Hnro	Etunimi	Sukunimi	Syntymävuosi	Aloituspvm	Yrtunnus
1	Mikko	Mikkonen	1965	1.2.1999	111
2	Tiina	Taitava	1970	12.3.2000	222
3	Riina	Rovasti	1980	30.5.2002	121
4	Liisa	Leppänen	1963	2.3.1990	121
5	Pekka	Ponteva	1985	1.6.2004	122
6	Raili	Tenhunen	1961	2.6.1985	225

Kuva 6: Esimerkki yhden suhde yhteen –yhteydestä (Hiltunen)

Relaatioalgebran neljä perusoperaatiota ovat:

- Liitos eli join

  Liitoksessa kaksi taulua yhdistetään uudeksi tauluksi avaimen perusteella. Uudessa taulussa on sarakkeita yhteensä sama määrä kuin alkuperäisissä yhteensä - 1 ja rivien määrä on sama kuin siinä taulussa, jossa rivejä oli enemmän. Inner join –liitoksessa syntyvässä taulussa on vain sellaisia rivejä, joiden tiedot löytyvät mo-

lemmista tauluista. Outer join –liitoksessa syntyvään tauluun tuodaan alkuperäisistä tauluista myös sellaisia rivejä, joille ei löydy vastinparia toisesta taulusta.

- Yhdiste eli union

  Yhdisteessä kaksi taulua joissa on samanlaiset sarakkeet yhdistetään uudeksi tauluksi. Yhdisteessä syntyvässä taulussa rivien määrä on sama kuin niiden rivien määrä jotka olivat identtisiä kahdessa alkuperäisessä yhteensä.

- Valinta eli selection

  Valinnassa taulusta valitaan uuteen tauluun vain tietyt rivit. Tällä operaatiolla tuotetaan relaatiotietokannoista esille kulloinkin haluttujen ehtojen mukaiset rivit.

- Projektio

  Projektiossa taulusta valitaan uuteen tauluun vain tietyt sarakkeet. Tätä operaatiota käytetään usein tietokannan normalisoinnissa.

Feuerlichtin mukaan relaatiotietokannat ovat harvinainen esimerkki sellaisesta tilanteesta, jossa teoreettinen malli on edeltänyt ja ohjannut teknologista implementointia. Feuerlicht myös kuvaa, miten relaatiotietokannat ratkaisivat kaksi toisistaan riippumatonta aikaisempien tietomallien ongelmaa: ne mahdollistivat sovelluslogiikkaohjelmien irrottamisen tietokannasta koska ne tarjosivat tehokkaan tuen datan itsenäisyydelle, ja ne vapauttivat tietokantasovellusten kehittäjät ohjelmallisesti navigoinnista tietoon esittelemällä ei-proseduraalisen kyselykielen. (Feuerlicht 2010, 165)

8.2.1 SQL-kieli

Kyselykielet on perinteisesti suunnattu yhdelle tietomallille, jolloin kyselykielellä tehty kysely voidaan kohdistaa vain tähän tietomalliin pohjautuviin tietolähteisiin. SQL on alun perin suunniteltu relaatiotietokannoille, jotka pohjautuvat relaatiomalliin. Jos tietoa on tallennettuna eri muodoissa, tarvitaan myös useita kyselykieliä käsittelemään näitä tietoja. (Vainio 2012, 1) Aikaisemmilla tietomalleilla ei tyypillisesti ollut omia itsenäisiä kyselykieliä, vaan kyselyt upotettiin osaksi isäntäkieltä, kuten esim. COBOLia.

SQL-kieli on syntynyt IBM:n laboratoriossa rakennettaessa relaatiotietokannan prototyyppiä nimeltä System R vuosina 1972 -73. SQL on lyhenne termistä Structured English Query Language. SQL-kielen avulla pyrittiin esittämään relaatioalgebran matemaattisia notaatioita käyttäjäystävällisemmällä tavalla. Tässä onnistuttiin ilmeisen hyvin, sillä SQL-kieli

mielletään siinä määrin dominoivaksi relaatiotietokantojen käsittelykieleksi, että relaatiotietokannoista joskus käytetään termiä ”SQL-tietokannat”. SQL kielenä jaetaan yleensä kahteen osaan: Data Definition Language (DDL) jolla määritellään tietokannan rakenne, sekä Data Manipulation Language (DML) jolla käsitellään tietokannan sisältöä.

ANSI (American National Standards Institute) muodosti vuonna 1986 virallisen SQL-standardin, jonka ISO (International Organization for Standardization) ratifioi seuraavana vuonna. Sen jälkeen standardista on tullut uusia versioita. Kaikki tuotevalmistajat eivät kuitenkaan tue kaikki standardoituja ominaisuuksia ja tuotekohtaisia lisäominaisuuksia on määritelty eri tietokantatuotteissa käytettäviin kieliversioihin.

8.3 Oliotietovarastot

Olio-ohjelmointiparadigman myötä alettiin kehittää oliotietovarastoja. Petri Kotirannan mukaan oliotietokannoissa lähestymistapa datan organisointiin muistuttaa olio-ohjelmointia. Oliotietokannassa tietomalli nähdään joukkona olioita. Olioiden tietosisältö on tallennettu attribuutteihin ja attribuuteilla on niitä käsittelevät metodit. Oliotietokannat eivät ole yleistyneet, vaikka niitä käytetäänkin esim. CAD-mallinnuksessa. Oliotietokannoille on pyritty kehittämään standardoitua käsittelymallia (ODMG), mutta se ei ole yleistynyt. Relaatiotietovaraston yleiset ominaisuudet ovat riittäneet useimmissa tapauksissa myös olio-ohjelmoinnin tarpeisiin, joten oliotietovarastoille ei ole ollut riittävän suurta elinlokeroa tarjolla. (Kotiranta 2015, 10.)

8.4 XML-tietovarastot

XML on SGML-kielestä (Standard Generalized Markup Language) johdettu merkkauskieli. XML-kielestä on muodostunut merkkauskielien de facto –standardi, ja sitä käytetään laajalti datan kuvaamiseen esim. Web Services –teknologioissa. 1990-luvulla kehitettiin paljon XML-kieleen perustuvia tuotteita, kieliä ja työvälineitä. Tuolloin on kehitetty myös tietovarastoja, jotka käyttävät XML-kieltä natiivikielenä, ja joihin voi tallentaa XML-dokumentteja. XML-tietokantoja voi siten luonnehtia varhaisiksi dokumenttikannoiksi. XML-tietokannoista ei kuitenkaan muodostunut sellaista valtavirtateknologiaa, joka olisi voittanut suosiossa relaatiotietokannat. Niillä on kuitenkin markkinaosuutta esim. sisällönhallinnan sovelluksien tietovarastoina. XML-tieto tallennetaan yleensä tekstimuodossa. XML-tietovarastojen käsittelykieleksi on vakiintunut XQuery.

8.5 NoSQL-tietovarastot

Big Data ja pilvipalvelut ovat esittäneet tietovarastoille haasteita, joihin vastaamiseksi on 2000-luvulla kehitetty uuden tyyppisiä tietovarastoja, joita kutsutaan yhteisnimikkeellä NoSQL-tietovarastot. NoSQL –tietovarastot ovat tietovarastotyyppien uusimpia tulokkaita eikä niiden käsittelyyn ole ainakaan vielä muodostunut minkään tietyn kielen hegemoniaa.

Digitaalisen tiedon määrä maailmassa lisääntyy jatkuvasti, ja lisääntymisen vauhti on tähän mennessä vain kiihtynyt. Tiedon määrän lisäksi myös tiedon vauhti lisääntyy jatkuvasti. Erilaiset anturit tuottavat jatkuvaa tietovirtaa esim. laitteiden toiminnasta tai henkilöiden elintoiminnoista. Tieto myöskin liikkuu nykyään jatkuvasti. Ennen tietoverkkoja tieto oli useimmiten paikallaan tietovarastossa, josta se noudettiin yksittäistä tiedonkäsittelytarvetta varten. Tietoverkot ovat luoneet uudenlaisen maailman; tieto liikkuu verkossa niin nopeasti, että ihmisaivot mieltävät tiedon olevan samanaikaisesti joka puolella verkkoa. Digitaalinen, saatavilla ja käytettävissä oleva tieto on myös muodoltaan moninaista. Aiemmin on usein mielletty, että tiedon käytettävyys edellyttää tiedon strukturointia. Joihinkin käyttötarkoituksiin tiedon tulee edelleenkin olla strukturoitua, mutta tällainen mielikuva on johtunut osittain myös käytettävissä olevien tekniikoiden puutteista – strukturoimattoman tiedon käsittelyyn ei ole ollut käytännöllisiä välineitä.

Suuret kaupalliset toimijat, joista Google usein mainitaan edelläkävijänä, ovat 2000-luvulla keränneet huomattavia tietomassoja. Näiden tietomassojen käsittelyyn on täytynyt kehittää uudenlaisia välineitä.

NoSQL –tietovarastot on kehitetty vastaamaan toisenlaisiin haasteisiin kuin relaatiotietovarastot. Strauch kuvaa luentomonisteessaan NoSQL-liikettä, joka syntyi 2000-luvun alkupuolella. Termiä NoSQL käytettiin ensimmäisen kerran vuonna 1998 kuvaamaan relaatiotietovarastoa, jonka käsittelyssä ei ollut välttämätöntä käyttää SOL-kieltä. Termi otettiin uudelleen käyttöön vuonna 2009 kuvaamaan tietovarastoja, joissa käytettiin jotain muuta tietomallia kuin relaatiota. (Strauch & Kriha (lecturer) 2011, 2) Eric Lai kuvaa artikkelissaan kesäkuussa 2009 pidettyä seminaaria, jonka osallistujat ”olivat tulleet kertomaan kuinka he olivat kukistaneet hitaiden, kalliiden relaatiotietokantojen tyrannian tehokkaammilla ja halvemmilla tiedonhallintatavoilla” (Lai 2009).

NoSQL-tietovarastoilla ei ole muuta varsinaista yhteistä nimittäjää kuin se, että ne eivät ole relaatiotietokantoja. NoSQL ei viittaa SQL-kielen kieltämiseen vaan pikemminkin siihen, että NoSQL-tietovarastoa ja sen sisältämää tietoa on mahdollista käsitellä myös

muulla kielellä kuin SQL-kielellä. Cattellin mukaan termi ”NoSQL” viittaa käsitteisiin ”Not Only SQL” tai ”Not relational”, mutta itse termillä ei ole yleisesti hyväksyttyä määritelmää (Cattell). Päärnin mukaan NoSQL-tuotteet eivät käytä SQL-kieltä kyselykielenään, mutta SQL-kieltä saatetaan käyttää osana tietovaraston hallintajärjestelmää. (Päärni, 14)

Cattell kuvaa NoSQL-tietovarastoille tyypillisiä ominaisuuksia seuraavasti: NoSQL-systeemien tyypillinen piirre on tiedon skaalautuvuus horisontaalisesti useille palvelimille. Tämä mahdollistaa suuren määrän yksinkertaisia luku/kirjoitus-operaatioita ajanjaksossa. Tällaista yksinkertaista operaatiokuormaa kutsutaan perinteisesti OLTP:ksi (online transaction processing), mutta se on tavanomaista myös nykyaikaisissa web-sovelluksissa. NoSQL-systeemit eivät yleensä tarjoa ACID:in mukaisia tapahtumankäsittelyominaisuuksia: tallennukset/päivitykset leviävät kaikille palvelimille aikanaan, mutta CAP-teoreemassa tarkoitettua eheyttä (consistency) ei voida taata. Onkin ehdotettu, että NoSQL-systeemeissä BASE-ominaisuudet korvaavat ACID-ominaisuudet. Perusajatuksena on se, että ACID-rajoitteista luopumalla saavutetaan parempi suorituskyky ja skaalautuvuus (Cattell). Skaalautuvuus saavutetaan NoSQL-tietovarastoissa usein n.s. sharding–tekniikalla (Sharding Introduction). Ari Hovin kuvauksen mukaan nämä tietokannat ovat tyypillisesti tinkineet joistakin relaatiokantojen tapahtumankäsittelyn eheys- tai turvallisuusominaisuuksista suorituskyvyn ja laajennettavuuden vuoksi. Tekstien, kuvien ja muun ei-strukturoidun tiedon tuki on parempi kuin relaatiokannoissa. (Hovi 2015)

Eräs yleispiirre NoSQL-tietovarastoissa on skeemattomuus tai joustava (flexible) skeema. Relaatiokannoissa taulujen rakenteet eli skeema määritellään ennenkuin tietoja voidaan tallettaa. NoSQL –kannat ovat yleensä ns. skeemattomia tai niiden skeema on joustava, eli rakennetta ei määritellä etukäteen ainakaan kovin tarkalle tasolle saakka. Koska tietovaraston rakennetta ei tarvitse määritellä etukäteen, skeemattomuus nopeuttaa tietovaraston rakentamisvaihetta. Skeemattomaan/joustavaskeemaiseen tietovarastoon on helppo lisätä dataa jolla on dynaaminen rakenne. Skeemattomuus mahdollistaa suurien tietomäärien tallentamisen nopeilla kirjoitusoperaatioilla, mikä sekin tukee Big Data –lähestymistapaa. Tietovaraston rakenne täytyy kuitenkin tuntea tietojen lukuvaiheessa. Skeemattomuus tavallaan siirtää monimutkaisten toimenpiteiden suoritusajankohtaa; tallentaminen on yksinkertaisempaa ja haku monimutkaisempaa kuin relaatiotietovarastoissa. Tiedon hakua tietovarastosta voidaan toteuttaa käyttäjien tai sovelluslogiikan tekemänä. Molemmissa tapauksissa NoSQL-tietovarastosta tietoa hakevien käyttäjien/sovellusten täytyy pystyä tulkitsemaan tiedon rakenne. Jos tietovarastosta haetaan tietoa ohjelmallisesti ja tiedon rakennetta ei ole kuvattu tietokantaskeemana, rakenne täytyy kuvata tietoa hakevaan ohjelmaan. Näin NoSQL –tietovarastot lähestyvät eräällä ta-

valla relaatiotietokantoja edeltänyttä aikaa, jolloin tiedon käsittely ei vielä ollut eriytynyt muusta ohjelmakoodista.

NoSQL-tietovarastot eivät kuitenkaan ole ominaisuuksiltaan keskenään samanlaisia. Koska NoSQL-tietovarastoilla ei ole yleisesti hyväksyttyä määritelmää, niillä ei myöskään ole yhtenäistä luokittelua. Seuraavissa luvuissa kuvataan muutamia erilaisia NoSQL-tietovarastotyyppejä, jotka on luokiteltu tietomallin mukaan . Luokitteluna on käytetty Soikkelin kuvaamaa luokittelua, joka on esitetty kuvassa 7.

NoSQL-tietokantojen luokittelu tietomallin mukaan

Tietokanta	Tietomalli	Käyttökohteet
Redis Memcached SimpleDB DynamoDB Riak	Avain-arvo-pari	välimuisti tiedon hajatus jonot reaaliaikaiset palvelut
MongoDB Amazon CouchDB	Dokumentti	yleiskäyttö
Cassandra Hadoop Hypertable Base SimpleDB	Sarake	tietovarastointi asiakkuudenhallinta analytiikka
Neo4j Infinite Graph TITAN AllegroGraph	Verkko	sosiaalinen media tieteellinen tilastointi sisällönhallinta suosittelujärjestelmät

Taulukko 2: NoSQL-tietokantojen luokittelu tietomallin ja käyttökohteiden mukaan

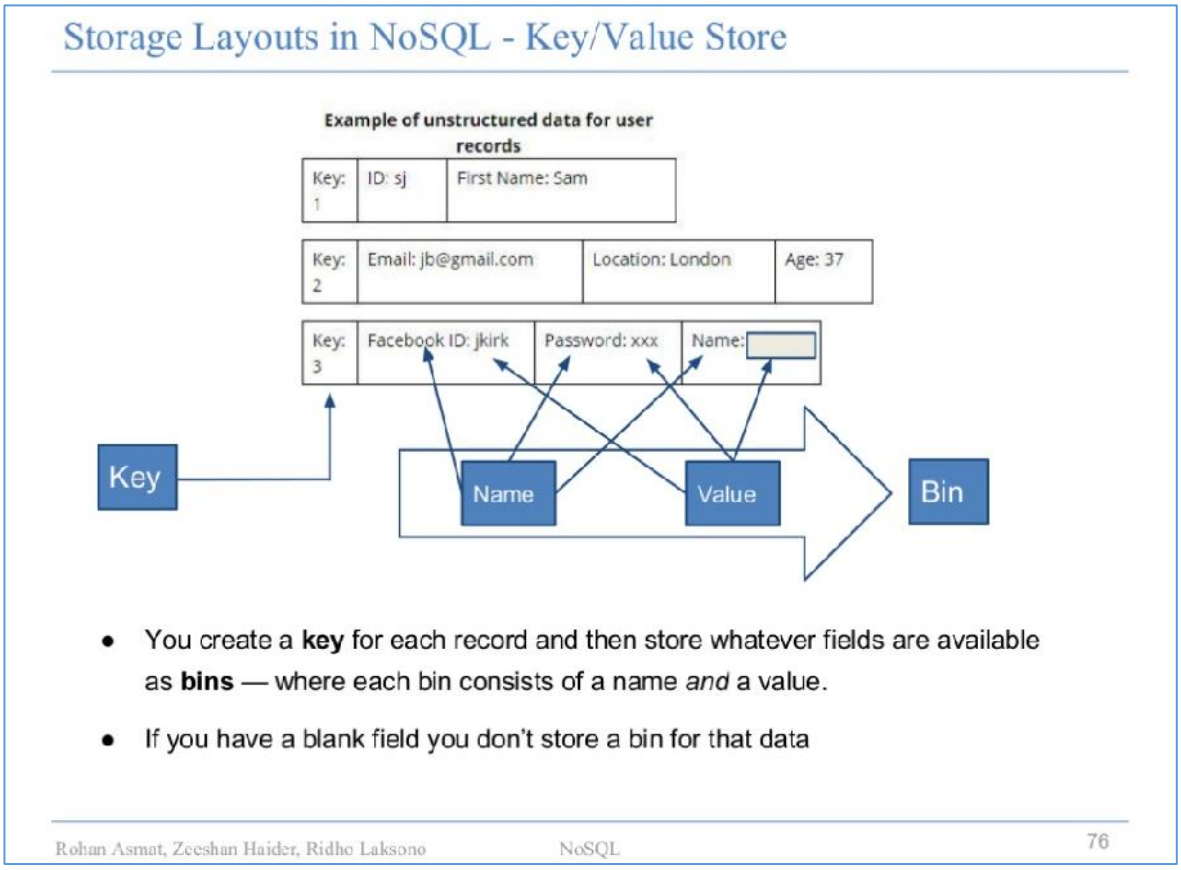
Kuva 7: NoSQL –tietokantojen luokittelu tietomallin ja käyttökohteiden mukaan (Soikkeli 2015, 19)

8.5.1 Avain-arvo-varastot

Avain-arvo-varastoille on yhteistä yksinkertainen tietomalli. Näissä tietovarastoissa tietomallina on matriisin tai hakemiston kaltainen rakenne, jossa tietojen haku ja tallennus tapahtuu avaimen kautta. Strauchin mukaan avain-arvo-varastoja on ollut käytössä jo kauan, mutta viime vuosien aikana tämän tyyppisiä tietovarastoja on tullut runsaasti lisää. Moderneissa avain-arvo-varastoissa skaalautuvuutta suositaan CAP-teoreemassa tarkoitetun eheyden kustannuksella. Nämä tietovarastot tarjoavat tällöin ominaisuuksia, jotka ovat hyödyksi esim. ad-hoc-kyselyissä ja analytiikassa. (Strauch & Kriha (lecturer) 2011,

52) Sandborgin mukaan avain-arvo-varastojen arvoilla ei yleensä ole mitään tietokantajärjestelmälle näkyvää semantiikkaa. Arvot ovat itsenäisiä ja eristettyjä toisistaan. Arvojen mahdollisia keskinäisiä suhteita on ylläpidettävä sovellusohjelmassa. Tietokannan rakenne on hyvin joustava. Eri tyyppisiä arvoja voidaan lisätä ilman suuria muutoksia olemassa olevaan tietokantaan. Avain-arvo-varastojen tietokanta onkin usein täysin skeematon. Tietorakenteen tietojen indeksointi ja hakutoiminnot perustuvat yksilöivään avaimeen. Avain-arvo-pareja voidaan yleensä myös ryhmitellä jollakin tavalla. (Sandborg 2012, 28).

Asmat, Malik ja Laksono ovat esittäneet avain-arvo-varaston tietomallin kuvallisesti seuraavalla tavalla:



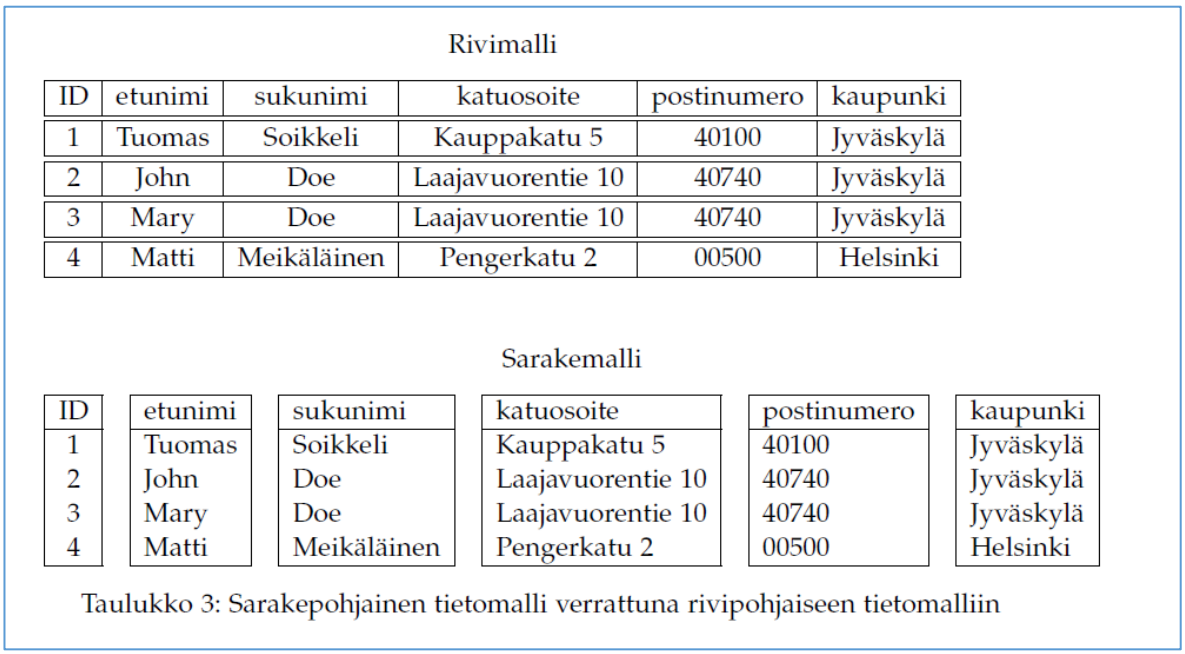
Kuva 8: Storage Layouts in NoSQL – Key/Value Store (Asmat, Malit & Laksono 2015, 76)

8.5.2 Sarakeperhevarastot

Soikkelin kuvauksen mukaan sarakepohjaiset tietokannat tallentavat dataa relaatiomallin tavalla tauluihin, mutta sarakepohjaisesti. Relaatiomallissa data tallennetaan rivien mukaan. (Soikkeli 2015, 22) Sandborg lisää tietomalliin myös aikaulottuvuuden: ”Sarakeperhevarastojen tietomallissa on yleensä myös kolmas ulottuvuus, nimittäin aika. Näiden tietokantajärjestelmien tauluihin voidaan tallentaa samasta tiedosta useita eri versioita. Periaatteena on, että kun tietoja muokataan, siitä tallennetaan aina uusi versio. Vanha versio jää myös talteen. Nämä versiot indeksoidaan sekä erotellaan toisistaan aikaleiman (times-

tamp) perusteella. Sarakeperheiden tietomallia voidaan kutsua rivien, sarakkeiden ja aikaleimojen takia moniulotteiseksi.” (Sandborg 2012, 28)

Soikkeli kuvaa rivimallin ja sarakemallin eroa kuvan 9 avulla:



Rivimalli

ID	etunimi	sukunimi	katuosoite	postinumero	kaupunki
1	Tuomas	Soikkeli	Kauppakatu 5	40100	Jyväskylä
2	John	Doe	Laajavuorentie 10	40740	Jyväskylä
3	Mary	Doe	Laajavuorentie 10	40740	Jyväskylä
4	Matti	Meikäläinen	Pengerkatu 2	00500	Helsinki

Sarakemalli

ID	etunimi	sukunimi	katuosoite	postinumero	kaupunki
1	Tuomas	Soikkeli	Kauppakatu 5	40100	Jyväskylä
2	John	Doe	Laajavuorentie 10	40740	Jyväskylä
3	Mary	Doe	Laajavuorentie 10	40740	Jyväskylä
4	Matti	Meikäläinen	Pengerkatu 2	00500	Helsinki

Taulukko 3: Sarakepohjainen tietomalli verrattuna rivipohjaiseen tietomalliin

Kuva 9: Sarakepohjainen tietomalli verrattuna rivipohjaiseen tietomalliin (Soikkeli 2015, 23)

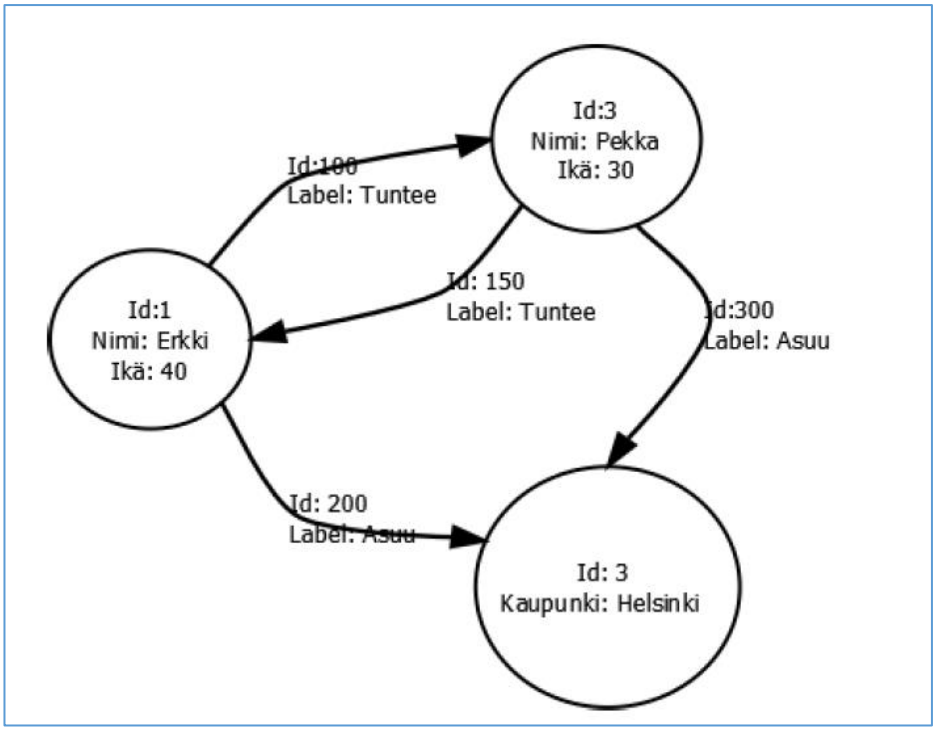
8.5.3 Dokumenttivarastot

Dokumenttivarastot ovat avain-arvo-pareista koostuvien dokumenttien kokoelmia, joissa avain-arvo-parin avain on yksilöivä tunnus dokumentin sisäisesti. Sen lisäksi jokaisella dokumentilla on oma yksilöllinen tunnus. Soikkelin mukaan dokumenttivarastot tarjoavat kattavan tavan tallentaa tiedon valtavaa massaa eivätkä ne keskity korkeisiin luku- ja kirjoitusnopeuksiin. Avain-arvo-parivarastoista eroten dataa voidaan hakea myös avaimen lisäksi arvoista, joten dokumenttivarastojen hakuominaisuudet ovat monipuoliset. Dokumenttivarastot ovat siis NoSQL-aatteen yleiskäyttöisiä tietovarastoja. Ne ovat jokseenkin tehokkaita, mutta niillä on myös piirteitä relaatiotietokannan ominaisuuksista. Dokumenttivarastoilla voi olla esimerkiksi monipuolinen hakukieli. (Soikkeli 2015, 20-21.) Dokumenttivarastoihin voi tallentaa tietoa erilaisissa formaateissa, esim. XML- tai JSON – formaatissa. Kirjoitus- ja lukuoperaatiot voivat nopeutua, jos tallennus- ja hakuvaiheissa ei tarvita formaatin muuntamista. Formaatin muuntamista ei tarvita silloin jos tietovarastosta/tietovarastoon siirrettävä tieto on jo valmiiksi XML- tai JSON-formaatissa. Molempia e.m. formaatteja käytetään nykyään yleisesti tiedonsiirrossa. Dokumentti voi olla hierarkkinen, jolloin monen taulun asiat voidaan tallettaa yhteen dokumenttiin.

8.5.4 Verkko- l. graafitietovarastot

Soikkeli kuvaa verkkotietovarastoja seuraavasti: ”Verkkotietokannoissa tieto tallennetaan verkkomallia käyttäen. Verkkomalli esittää datan verkon solmuina ja niiden välisinä suhteina. (Vicknair ym., 2010) Verkko eli graafi on tietojenkäsittelytieteissä tietomalli, joka koostuu joukosta yksittäisiä solmuja sekä niiden välillä olevista kaarista. Nämä kaaret kuvaavat verkon solmujen välisiä suhteita. Verkkotietokannat sopivat käyttötarkoituksiin missä tieto on useassa relaatiossa. […] Relaatiotietokannan useat JOIN-operaatiot ovat raskaita operaatioita, verkkotietokannassa ne ovat tavallinen operaatio johtuen tietomallista. Useita relaatioita, eli verkon kaaria sisältävä skeema on myös helposti hajautettavissa niin myöskin koostettavissa MapReducen avulla.” (Soikkeli 2015, 24) Verkkotietovarastojen tyypillisiä käyttökohteita ovat siten tilanteet, joissa halutaan tallentaa tietoa verkostoista, kuten esim. sosiaalisen median verkostoista.

Kotiranta on kuvannut graafitietovaraston tietomallia kuvassa 10:



Kuva 10: Graafitietokannan tietomalli (Kotiranta 2015, 17)

Tässä kuvassa tietueet esitetään kolmena solmuna, jotka tässä tapauksessa ovat kaksi ihmistä ja kaupunki. Tietueiden välisiä relaatioita ilmaisevat solmujen väliset kaaret, jotka tässä tapauksessa ilmaisevat ihmisten väliset suhteet toisiinsa ja missä kaupungissa he asuvat. (Kotiranta 2015, 17)

9 Tietovarastoratkaisun valintaan vaikuttavat tekijät

Eri tyyppisillä tietovarastoratkaisuilla ja eri tietomalleihin pohjautuvilla tietovarastoratkaisuilla on erilaisia ominaisuuksia. Organisaation on selvitettävä millaiseen käyttöön tietovarastoa tarvitaan. Jos on tarvetta tapahtumien käsittelyyn, voidaan pohtia hoidetaanko tapahtumien hallinta tietovaraston tasolla vai tietovaraston tietoja käyttävien sovellusten logiikassa. Relaatiotietokantojen hallintajärjestelmät sisältävät yleensä tapahtuman hallintaominaisuuksia, jolloin niitä ei tarvitse toteuttaa sovelluslogiikkaan. Muut ratkaisun valintakriteerit johtuvat pitkälti tietovaraston ulkopuolisista seikoista. Tietohallintostrategia, teknologian kypsyys, osaamisen saatavuus, markkinatilanne, liiketoimintatarpeet ja muu kokonaisarkkitehtuuri sekä käytettävät tiedonsiirtoformaatit ovat kaikki asioita jotka vaikuttavat tietovarastoratkaisun valintaan. Organisaation keskeisimpänä tehtävänä lieneekin tunnistaa näiden elementtien lähtökohdat, painopisteet ja kehittämistarpeet, sekä se osaaminen jota tarvitaan näiden elementtien hahmottamiseen kulloisessakin ratkaisutilanteessa.

9.1 Tietohallintostrategia

Organisaation tietohallintostrategiasta riippuu miten suuri osa tiedon käsittelyyn tarkoitetuista järjestelmistä tulisi olla omaa tuotantoa ja miten suuri osa hankittua. Osana strategiaa mietitään myös, millaiset järjestelmät tai miten suuri osa järjestelmistä halutaan hankkia valmiina tuotteina. Yksiselitteistä vastausta näihin kysymyksiin ei ole.

9.2 Teknologian kypsyys

Riippumatta siitä tuotetaanko tietojärjestelmiä omana tuotantona vai ostetaanko niitä organisaation käyttöön markkinoilta, eräs huomioon otettava näkökulma on teknologian kypsyys. Uudet teknologiat tarjoavat usein kaivattuja ratkaisuja olemassa oleviin, tunnistettuihin ongelmiin. Uuden teknologian kypsyys saattaa kuitenkin olla riittämätön organisaation tarpeita ajatellen. Uusissa ratkaisuissa saattaa olla virheitä ja heikkouksia, joita ei ole vielä ehditty havaita. Vaikka virheitä ei olisikaan, uudet ratkaisut eivät vielä elinkaarensa alkuvaiheessa ole yleensä käytettävyydeltään parhaita mahdollisia. Kun uusi teknologia havaitaan käyttökelpoiseksi, sen ympärille alkaa syntyä ekosysteemi. Syntyy kieliä, kehitysohjelmistoja, apuvälineitä, menetelmistöjä sekä adaptereita jotka sovittavat uutta teknologiaa yhteen olemassa olevien teknologioiden ja tuotteiden kanssa.. Uusinta uutta varten ei ole vielä ehditty kehittää apuvälineitä. Uudet teknologiat ovatkin tyypillisesti koo-

dareitten valtakuntaa, jossa toiminnallisuuksia toteutetaan jollain koodauskielellä ja word-padilla – uudelle teknologialle ei ole vielä ehditty toteuttaa debuggausohjelmistoja. Tämän vuoksi uusien teknologioiden hyödyntäminen saattaa olla virhealttiimpaa kuin valtavirta-ratkaisujen hyödyntäminen.

Erityisesti viranomaistoiminnassa järjestelmien toimintavarmuus on tärkeä näkökohta. Toimintavarmuuden varmistamiseksi viranomaistoiminnassa on erityisen tarkkaan harkittavam millä tavalla uusia teknologioita otetaan käyttöön. Monissa tapauksissa pilotointi on toimiva ratkaisu; uutta teknologiaa ei oteta käyttöön heti kaikkialla, vaan kokeillaan sen ominaisuuksia rajatusti. Pilottikäytössä ei yleensä kokeilla kriittistä toiminnallisuutta, vaan jotain sellaista toiminnallisuutta, jossa riskit ovat paremmin hallittavissa. Pilottikäytön rajaus voidaan tehdä vaikkapa kohdejoukon perusteella siten, että kokeilukäyttöön osallistuu vain tietty käyttäjäryhmä/asiakasryhmä tai tietynlainen otos jostain ryhmästä. Muita mahdollisia rajauksia ovat esim. ajallinen rajaus tai käyttötilannerajaus. Käyttöönotto voidaan toteuttaa joko pilotointina tai kokonaisuuden rinnakkaiskäyttöönottona. Tällöin olemassa olevaa ratkaisua ja käyttöönotettavaa ratkaisua käytetään rinnakkain ennalta päätetyn ajanjakson ajan. Olemassa oleva ratkaisu toimii tuotantoratkaisuna, kunnes rinnakkaiskäytöllä on voitu riittävän luotettavasti todeta se, että uusi ratkaisu on toimiva ja tuottaa saman lopputuloksen kuin olemassa oleva tuotantoratkaisu.

9.3 Osaamisen saatavuus

Uusien teknologiaratkaisujen hankintaa harkittaessa organisaation kannattaa tutkia erityisen huolellisesti osaamisen saatavuutta. Toimintavarmuus edellyttää, että osaamista on saatavilla. Osaamista täytyy olla saatavilla riippumatta siitä ostetaanko ratkaisu organisaation ulkopuolelta vai tuotetaanko se itse. Osaamisen saatavuus vaikuttaa hankintahintaan, mutta erityisesti sillä on vaikutusta hankinnan elinkaaren aikaisiin kokonaiskustannuksiin. Uusien teknologioiden osaajia ei teknologian elinkaaren alkuvaiheessa ole aina riittävästi saatavilla. Toisaalta poistumassa olevien teknologioiden osaajia voi olla saatavilla vielä vähemmän ja siten vielä kalliimmalla hinnalla. Osaamisen hallinnan kannalta organisaation turvallisin vaihtoehto vaikuttaisi olevan valtavirtateknologioiden käyttö ja teknologioiden uusiminen sellaisella aikataululla, että niitä ei organisaatiossa käytetä aivan niiden elinkaarien viimeisille hetkille saakka.

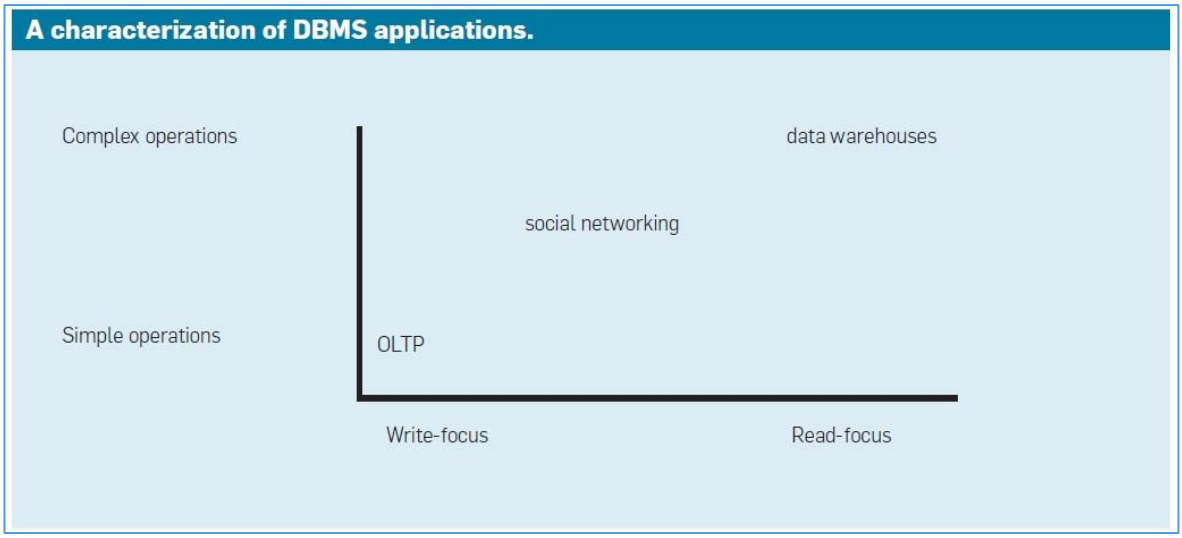
9.4 Valmisohjelmistot

Tietojärjestelmätuotteissa eli valmisohjelmistoissa ei aina ole mahdollisuutta valita erilaisia tietovarastoratkaisuja. Valmisohjelmistot yleensä tehty tietynlaiseen käyttötarkoitukseen,

joten käytettävä tietovarastoratkaisu on valittu sen mukaisesti. Valmisohjelmistoissakin saattaa olla mahdollisuus käyttää eri tietokantatuotteita, mutta useimmiten nämä tietokantatuotteet kuitenkin kuuluvat samaan kategoriaan, esim. relaatiotietovarastoihin.

9.5 Markkinatilanne

Stonebrakerin ja Cattellin mukaan 1970- ja 1980 –luvuilla kaupallisilla markkinoilla oli olemassa vain yksi suuri yhtenäinen tietovarastotyyppi, eli liiketoiminnan operatiiviseen tietojenkäsittelyyn (OLTP) tarkoitetut tietovarastot. Sen jälkeen tietovarastojen käyttökohteita ja sitä myötä valikoimaa on tullut lisää, esim. data warehouset, tieteelliseen käyttöön tarkoitetut tietovarastot, sosiaalisen median sovellukset ja pelisovellukset. (Stonebraker & Cattell 2011, 72). Kirjoituksessaan Stonebraker & Cattell esittävät tietovarastomarkkinat käyttötarkoituksen mukaisena nelikenttänä (kuva 11). Tässä kuvassa horisontaaliakseli kuvaa luku-/kirjoitusoperaatiokeskeisyyttä. Vertikaaliakseli kuvaa operaatioiden monimutkaisuutta – onko tyypillinen käyttötapa yksittäisten tietojen luku/kirjoitus, vai luetaanko/kirjoitetaanko tyypillisesti tuhansia tietoja kerralla.



Kuva 11: A characterization of DBMS applications (Stonebraker & Cattell 2011, 74)

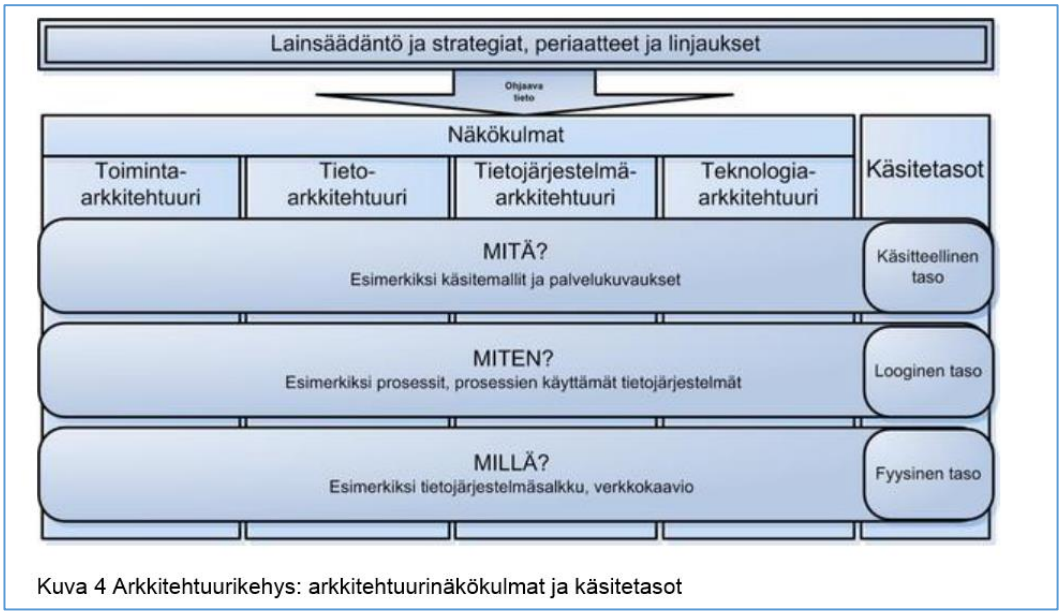
Stonebraker ja Cattell kuvaavat tietovarastoja niiden tyypillisten käyttötapojen mukaan seuraavasti: Perinteiset, liiketoiminnan operatiiviseen tietojenkäsittelyyn tarkoitetut tietovarastot ovat kirjoitus-painotteisia. Data warehouse –tyyppiseen tietojenkäsittelyyn tarkoitetut tietovarastot ovat luku-painotteisia. Monien tietovarastoratkaisujen käyttötapa on jotain tältä väliltä; esimerkiksi sosiaalisen median tietovarastojen dataan kohdistuu enimmäkseen yksinkertaisia operaatioita, ja tietovaraston käyttötapaan kuuluu sekä kirjoitus- että lukuoperaatioita. (Stonebraker & Cattell 2011, 73-74.)

9.6 Kokonaisarkkitehtuuri

Tietovarastoratkaisu ei aina ole keskeinen tekijä ICT-ratkaisun valintatilanteessa. Jos tietovarastoratkaisu voidaan valita, keskeisenä tekijänä ovat liiketoiminnalliset tarpeet. ICT-ratkaisut palvelevat liiketoiminnallisia tarpeita, ja tämä pätee myös tietovarastoratkaisuihin. Stonebrakerin ja Cattellin jaottelu tietovarastojen operaatiotyypeistä ja kirjoitus-/lukusuuntautuneisuudesta (kuva 11) saattaa olla avuksi silloin kun hahmotetaan millaisiin liiketoiminnallisiin tarpeisiin tietovarastoa eniten tarvitaan.

Tietovarastoratkaisun valinta voi harvoin olla myöskään erillinen päätös, vaan valintatilanteessa on otettava huomioon laajempi tietojen käsittelyn kokonaisuus. Kokonaisarkkitehtuuri on menetelmä, jonka avulla kokonaisuutta on mahdollista hallita ja kehittää systemaattisesti. Suomessa tuli 1.9.2011 voimaan Laki julkisen hallinnon tietohallinnosta eli Tietohallintolaki. Lain tarkoituksena on tehostaa julkisen hallinnon toimintaa sekä parantaa julkisia palveluja ja niiden saatavuutta säätämällä julkisen hallinnon tietohallinnon ohjauksesta ja tietojärjestelmien yhteentoimivuuden edistämisestä ja varmistamisesta (Tietohallintolaki 1§). Lain mukaan Valtiovarainministeriön tehtävänä on tämän lain mukainen julkisen hallinnon viranomaisten tietohallinnon yleinen ohjaus. Ministeriön tulee erityisesti huolehtia julkisen hallinnon toiminta-, tieto-, järjestelmä- ja teknologia-arkkitehtuurin (kokonaisarkkitehtuuri) suunnittelusta ja kuvaamisesta (Tietohallintolaki 4§ 1 mom). Myös tietovarastoratkaisu on sovitettava organisaation kokonaisarkkitehtuuriin, joten tietovarastoratkaisua ei voida ajatella erillisenä, muusta arkkitehtuurista riippumattomana päätöksenä.

Julkisen hallinnon tietohallinnon neuvottelukunnan eli JUHTA:n JHS-jaoston tehtävänä on Tietohallintolain 5.2 §:n nojalla valmistella julkisen hallinnon tietohallintoa koskevia julkisen hallinnon suosituksia. Jaosto on julkaissut mm. suosituksen JHS 179 ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen. Suositus on tämän tutkielman kirjoitushetkellä päivitettävänä. Suosituksessa määritellään menetelmä, jolla organisaation kokonaisarkkitehtuuri suunnitellaan sekä annetaan suositukset kokonaisarkkitehtuurin eri osa-alueiden kuvausten laatimisesta. Suosituksen tarkoituksena on antaa yhtenäinen suunnittelumenetelmä, suunnittelun viitekehys sekä yhtenäiset kuvaustavat ja -mallit julkisen hallinnon organisaatioiden kokonaisarkkitehtuurin kehittämiseen sen eri vaiheissa. Suosituksen arkkitehtuurikehys sisältää neljä eri arkkitehtuurinäkökulmaa: toiminta- , tieto-, tietojärjestelmä- ja teknologia-arkkitehtuurinäkökulman. Suosituksen arkkitehtuurikehys pohjautuu TOGAF-mallin arkkitehtuurikehykseen. Arkkitehtuurikehystä kuvataan suosituksessa seuraavan kuvan avulla:



Kuva 4 Arkkitehtuurikehys: arkkitehtuurinäkökulmat ja käsitetasot

Kuva 12: Arkkitehtuurikehys: arkkitehtuurinäkökulmat ja käsitetasot (JHS 179 ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen, s. 10)

9.7 Tietovaraston ja sovellusten välinen työnjako

Organisaation tulee harkita tarvitaanko tietovarastoa tapahtumankäsittelyyn. Relaatiotietokannat ja niiden hallintajärjestelmät on optimoitu tämän tyyppisiä käsittelyjä varten. Päärnin mukaan NoSQL-tietovarastoissa perinteisen relaatiotietokannan hallintajärjestelmien tapahtumanhallintaominaisuuksista on osittain luovuttu tiedon paremman saatavuuden kustannuksella. Monille Internet-sovelluksille, kuten pankki- ja finanssialan sovelluksille, takeet tiedon eheydestä ovat pakollisia tietokannan hallintajärjestelmän ominaisuuksia. (Päärni, 7.)

On kuitenkin huomattava, että tapahtumanhallintaa ei tarvitse välttämättä toteuttaa tietovaraston ominaisuuksilla. Tapahtumanhallintaa voidaan toteuttaa myös sovelluslogiikan tasolla. Jos näin päätetään tehdä, tämän tulee olla tietoinen päätös ja sovelluslogiikka tulee rakentaa tämän arkkitehtuurin mukaisesti. Tällaisen arkkitehtuurin toteuttaminen saattaa olla työläämpää ja riskialttiimpaa kuin tapahtumanhallinnan toteuttaminen tietovarastotasolla. Yleistystä asiasta ei kuitenkaan voi tehdä, koska vaihtoehdon toteuttamiskelpoisuus riippuu halutusta toiminnallisuudesta ja sen yksinkertaisuudesta/monimutkaisuudesta. Stonebrakerin ja Cattellin kirjoituksessa kuvataan mitä on otettava huomioon kun halutaan rakentaa skaalautuvaa suorituskykyä n.s. ’Simple Operation’ –tietovarastoissa: sovellukset on suunniteltava skaalautuvuutta varten. Tämä tarkoittaa

mm. sovellusdatan jakamista sharding-tekniikalla sekä eheysvaatimusten painoarvon punnitsemista. (Stonebraker & Cattell 2011, 72)

9.8 Tietovirtojen formaatit

Eräänä näkökulmana tietovarastoratkaisun harkinnassa voi käyttää tietovirtojen formaatteja. Käytettävät tiedonsiirron formaatit kannattaa kartoittaa ennen tietovarastoratkaisun hankintaa.. Esim. joihinkin dokumenttitietovarastoihin voi tallentaa tietoa suoraan xml- tai json-formaatissa. Jos tietovarastoon tuodaan ulkoisista lähteistä tietoa pääasiassa näissä formaateissa tai sieltä on tarkoitus lähettää ulkoisille tahoille tietoa pääasiassa näissä formaateissa, saattaa olla hyödyllistä myös tallentaa tieto näissä formaateissa. Tällöin tallennus- ja/tai hakuvaiheessa säästytään tiedon formaattimuunnoksilta. Tämän tyyppisissä tietovarastoissa on mahdollista myös kohdistaa haku koko dokumentin sisältöön.

10 Pohdinta

10.1 Tutkimuksen johtopäätökset

Tämän tutkimuksen tutkimusongelmaksi kuvattiin: ”Mitä näkökulmia on huomioitava tietovarastoratkaisun valinnassa ja mitä uutta Big Data –lähestymistapa tuo mukanaan näihin näkökulmiin?” Tutkimuksen johtopäätökset saattavat vaikuttaa itsestään selviltä: valinnassa on otettava huomioon monia eri näkökulmia.

Tietovarastoratkaisu saattaa vaikuttaa tekniseltä yksityiskohdalta, joka ”vain täytyy olla olemassa”. Tietovarastot saatetaan joskus jopa nähdä esoteerisina ilmiöinä usein teknologiaan painottuvalla ICT-alalla. Tietojenkäsittelytieteessä tietokantojen, tietovarastojen ja hajautettujen järjestelmien kehitys on kuitenkin ollut tutkimuksen kohteena jo useita vuosikymmeniä, ja edistysaskeleet ovat saattaneet perustua matemaattisiin teorioihin. Big Data ilmiönä on tuonut tietojenkäsittelylle uudenlaisia haasteita, joihin myös tietovarastojen on kyettävä vastaamaan.

Tietovarastoratkaisu ei siten voi olla pelkkä tekninen yksityiskohta, eikä toisaalta myöskään erillinen saareke tietojenkäsittelyn kokonaisuudessa. Ratkaisun valinnassa on tärkeää tunnistaa kaikki ne näkökulmat jotka valintaan vaikuttavat tai voivat vaikuttaa. Vain kaikki tarpeelliset seikat huomioimalla on mahdollista tehdä hyviä päätöksiä. Tällainen lopputulos itsessään korostaa tietovarastojen roolia; päätöksentekoa varten tarvitaan riittävästi informaatiota, ja tietovarastojen eräs käyttökohde onkin päätöksentekoa varten tarvittavan datan kerääminen, säilyttäminen ja luovuttaminen – jotta päätöksentekoa varten tarvittavan informaation synty olisi mahdollista.

Tiedon käsittelyn tarpeet vaihtelevat saatavuuden, yksilöimisen/yksilöimättömyyden sekä kirjoitus-/lukuoperaatioiden painottumisen perusteella. Big Data ilmiönä on tuonut uudenlaisen lähestymistavan tiedon varastointiin ja tuonut uusia välineitä laajojen, vaihtelevien ja monimuotoisten tietomassojen käsittelyyn. Eri tyyppisillä tietovarastoratkaisuilla on erilaisia ominaisuuksia, jotka tukevat erilaisia tiedon käsittelytarpeita. Organisaation tulee harkita tarvitaanko tietovarastoa tapahtumankäsittelyyn. Relaatiotietokantojen hallintajärjestelmät sisältävät yleensä tapahtumien hallintaominaisuuksia. Jollei tietovarastoratkaisu sisällä tapahtumanhallinnan tukea ja tiedon käsittelytarve edellyttää sitä, tapahtumanhallinta täytyy toteuttaa sovelluslogiikan tasolla.

Tietovarastoratkaisun valintakriteereinä on otettava huomioon monia tietovaraston ulkopuolisia seikkoja: tietohallintostrategia, teknologian kypsyys, osaamisen saatavuus, mark-

kinatilanne, liiketoimintatarpeet ja muu kokonaisarkkitehtuuri sekä käytettävät tiedonsiirtoformaatit. Organisaation keskeisimpänä tehtävänä lieneekin tunnistaa näiden elementtien lähtökohdat ja kehittämistarpeet, sekä se osaaminen jota tarvitaan näiden elementtien hahmottamiseen kulloisessakin ratkaisutilanteessa.

10.2 Tutkimuksen hyödynnettävyys

Tutkimukseen on kerätty tietoa useista erilaisista tietovarastotyypeistä. Tavoitteena on ollut pitää tietovarastojen tyypittely riittävän yleisenä, jotta erilaiset tyyppiominaisuudet olisivat tunnistettavissa. Tutkimuksessa on mainittu joitain tuotenimiä, mutta tarjonta markkinoilla muuttuu jatkuvasti. Mainitut tuotenimet onkin tuotu esille vain esimerkkeinä kuvattavista tietovarastotyypeistä. Käytännön ratkaisutilanteissa on tutustuttava markkinatilanteeseen, jos valintatilanne on sellainen että tietovaraston valinnassa tulee kyseeseen valitseminen eri tuotteiden välillä. Näin ei asianlaita ole aina; joskus tietojärjestelmätuotteen valmistaja/myyjä on päättänyt tietovarastoratkaisun ostajan puolesta ja tarjoaa tuotteessaan vain yhtä vaihtoehtoa tai muutamaan keskenään samaan tietovarastotyyppiin kuuluvaa vaihtoehtoa.

Käytännön ratkaisutilanteita varten tutkimuksessa on tuotettu luettelo niistä seikoista, jotka vaikuttavat tietovarastoratkaisun valintaan. Kappaleessa 9 luettelo on kuvattu otsikkotason jaottelulla, joka on helppo tiivistää lyhyempään esitysmuotoon silloin kun tietovarastoratkaisuiden perusteluja esitellään organisaation johdolle.

10.3 Opinnäytetyöprosessi ja oma oppiminen

Tämä opinnäytetyö tehtiin Haaga-Helia ammattikorkeakoulun ohjeistuksen mukaisena työelämän kehittämistehtävänä projektimuodossa. Opinnäytetyöprosessissa oli siten käytössä projektinhallintamenetelmä sekä tutkimusmenetelmät. Projektinhallintamenetelmänä käytettiin toimeksiantajan menetelmistöön sisältyvää pienprojektin hallintamenettelyä. Projekti eteni alkuperäisen suunnitelman mukaisesti, joten projektin aikana ei esitetty yhtäkään muutospyyntöä. Projektin ohjausryhmä tuki projektipäällikköä projektin läpi viemisessä. Projektin tuotos oli siten suunnitellun laajuinen ja projekti päättyi suunnitellussa aikataulussa.

Tutkimustyö tähtäsi käytännön työvälineen luomiseen. Tutkimuksen tavoitteena oli tuottaa tietoa, jonka avulla on mahdollista tiivistää tietovarastoratkaisun valinnassa huomioon otettavat seikat kriteeristöksi. Työelämän kehittämistehtävänä oli tuottaa kriteeristö käy-

tännön apuvälineeksi tietovarastoratkaisujen valintatilanteissa. Tavoitteena oli siis tuottaa käytännön apuväline/menettely, jonka avulla voidaan esim. kilpailutustilanteissa valita oikea lähestymistapa ja tehdä perusteltu valinta erilaisten ratkaisujen välillä. Tutkimus oli luonteeltaan kvalitatiivinen tutkimus, jonka aineisto kerättiin kirjallista materiaalia hyödyntämällä ja havainnoimalla. Aineiston käsittelyssä korostuivat mm. sisällönanalyysi, mallien löytäminen ja tulkinta.

Tutkimustyön aihepiiri oli opinnäytetyön tekijälle osittain tuttu entuudestaan ja osittain vieras. Oman oppimisen kannalta katsottuna tämä saattaa olla paras mahdollinen lähtökohta tutkimuksen tekemiselle. Entuudestaan tutut asiat toimivat perustana, jolle uutta osaamista voi ryhtyä rakentamaan. Oppimisprosessissa pystyi selvästi huomaamaan erilaisia jaksoja ja erilaisia tapahtumia. Opinnäytetyöprojekti alkoi lähdemateriaaliin tutustumisella. Projektin edetessä lähdemateriaaliin palaaminen tuotti toisinaan äkillisiä oivalluksia, toisinaan asioiden oppiminen vaati kypsyttelyä. Oppiminen prosessina hahmottui projektin aikana selkeästi. Asiasisällön oppimisen lisäksi opinnäytetyöprojekti oli hyvä tapa opetella myös opiskelemista ja oppimista.

Lähteet

Abadi, D. J. 2009. Data management in the cloud: Limitations and opportunities. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. Luettavissa:

Asmat, M.R.A., Malik, Z.H. & Laksono, R. 12.2.2015. IIS – Seminar "NoSQL" Luettavissa: http://www.slideshare.net/rohanasmat/introduction-to-nosql-44600178. Luettu 16.11.2015.

Berman, J. J. 2013. Principles of Big Data. Preparing, Sharing, and Analyzing Complex Information. Elsevier Inc. USA.

Cattell, R. 2010. Scalable SQL and NoSQL data stores. ACM SIGMOD Record,39, 4 (December 2010), s. 12-27.

Feuerlicht, G. 2010. Database Trends and Directions: Current Challenges and Opportunities. Conference Paper: DATESO. April 2010, s. 163-174.

Garcia-Molina, H., Ullman J.D. & Widom J. 2009,2002. Database Systems. The Complete Book. Second Edition. Pearson Prentice Hall. USA.

Haaga-Helia 2010. Ammattikorkeakoulun opinnäytetyön sisältö ja menetelmät. Ohje. Haaga-Helia ammattikorkeakoulu.

Hiltunen, M. Tietokantasuunnittelun perusteet. Kurssimateriaali. Oulun seudun ammattiopisto. Kaukovainion yksikkö, liiketalous. Luettavissa: http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/tietokantasuunnittelun_perusteet/relaatiotietokannan_perusteet/perusteet.htm. Luettu 15.11.2015.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2009. Tutki ja kirjoita. 15. uudistettu painos. Tammi. Helsinki.

Hovi A. 2015. MongoDB haastaa relaatiotietokantoja. Blogikirjoitus. Luettavissa: http://www.arihovi.com/mongodb-haastaa-relaatiokantoja/?utm_source=lista1_210509&utm_campaign=f3808b7f41-

Syksy2015&utm_medium=email&utm_term=0_a675de72d6-f3808b7f41-325854573. Luettu 16.11.2015.

Iivonen, M. 2014. Skaalautuva verkkosovelluskehitys. Insinöörityö (AMK). Metropolia Ammattikorkeakoulu, Tietotekniikka.

JHS 179. Julkishallinnon suositus: ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen. Julkaisupäivä 8.2.2011. Luettavissa: http://www.jhs-suositukset.fi/suomi/jhs179. Luettu 14.11.2015.

Järvelin, K. ja Niemi, T. 1990. Hajautettujen faktatietokantojen käytön yksinkertaistaminen [Simplifying Retrieval from Distributed Fact Databases]. Kirjastotiede ja informatiikka 9 (1): 3-16, 1990.

Järveläinen,T. 2014.Tietomallien yhteensovittaminen siltahankkeessa. Liikenneviraston tutkimuksia ja selvityksiä 26/2014. Luettavissa: http://www2.liikennevirasto.fi/julkaisut/pdf8/lts_2014-26_tietomallien_yhteensovittaminen_web.pdf. Luettu 15.10.2015.

Kananen, J. 2009. Toimintatutkimus yritysten kehittämisessä. Jyväskylän ammattikorkeakoulu

Koistinaho, J. 2013. Korkean saatavuuden järjestelmän automaattinen testaus. Diplomityö. Tampereen Teknillinen Yliopisto. Tietotekniikan koulutusohjelma.

Kotiranta, P. 2015. Verkkotapahtumien keräysjärjestelmän tietokannan uudistaminen. Pro gradu –tutkielma. Tampereen yliopisto, Informaatiotieteiden yksikkö, Tietojenkäsittelyoppi

Lai E. 2009. Computerworld 6/2009: No to SQL? Anti-database movement gains steam. Luettavissa: http://www.computerworld.com/s/article/9135086/No_to_SQL_Anti_database_-movement_gains_steam. Luettu 9.11.2015

Laine H. 2000. Tietokantojen perusteet. Opetusmoniste. Helsingin yliopisto, Tietojenkäsittelytieteen laitos. Luettavissa: http://www.cs.helsinki.fi/u/laine/tikape/moniste/osa1.pdf. Luettu 16.8.2015.

Laine H. 2005. Tietokantojen perusteet, s 2005. Tietomallit. Opetusmoniste. Helsingin yliopisto/TKTL. Luettavissa:
https://www.cs.helsinki.fi/u/laine/tikape/s05/pdf/tietomalli_c.pdf. Luettu 16.8.2015.

Mayer-Schönberger, V. & Cukier, K. 2013. Big data : a revolution that will transform how we live, work and think. John Murray. Lontoo

Mustonen, T. 2003. Tapahtumankäsittely hajautetussa ympäristössä. Diplomityö. Lappeenrannan teknillinen yliopisto, Ttietotekniikka, tietoliikennetekniikan laitos

Nykänen, O. 1996. Älykkään opastusjärjestelmän toteuttaminen: verkostoon perustuva toteutus. Ohjelmistotekniikan seminaariesitelmä. Jyväskylän yliopisto, Matematiikan laitos. Luettavissa: http://www.mit.jyu.fi/opiskelu/seminaarit/bak/aopast/. Luettu 16.11.2015.

Ollikainen, V. Tiedonhallinnan perusteet. Verkko-oppimateriaali. Metropolia. Luettavissa: http://users.metropolia.fi/~olliv/Sovellusohjelmat/Materiaalit/Relaatiomalli.php. Luettu 10.11.2015.

Pritchett, D. 2008. BASE: An acid alternative. ACM Queue (May-June 2008), s. 48-55. Luettavissa: http://www.cs.cornell.edu/Courses/CS5412/2014sp/papers/p48-pritchett.pdf . Luettu 19.11.2015.

Päärni, A. 2012. Pilvisovelluksille sopivia tietokannan hallintajärjestelmiä. Kandidaatintutkielma. Jyväskylän yliopisto, tietojenkäsittelytieteiden laitos.

Sandborg, J. 2012. NoSQL-tietokannat. Graduseminaarityö . Luettavissa:
http://www.cs.helsinki.fi/u/paakki/semik12.html. Luettu 14.9.2015.

Savela, O. 2015. Helsinki. Minkä kokoinen on julkinen talous? Palkansaajien tutkimuslaitoksen raportti 30: Hyvinvointivaltio 2010-luvulla – mitä kello on lyönyt? (toim. Taimio Heikki), s. 14-34. Luettavissa: http://www.labour.fi/tutkimusjulkaisut/raportit/raportti30.pdf. Luettu 26.10.2015.

Sharding Introduction. MongoDB, Inc. Luettavissa:
https://docs.mongodb.org/manual/core/sharding-introduction/
Luettu 12.11.2015.

Silberschatz, A. , Korth, H. F. & Sudarshan, S. 2011. Database System Concepts, Sixth Edition, International Edition 2011. McGraw-Hill.

Sipilän hallituksen ohjelma. 2015. Luettavissa: http://vm.fi/documents/10623/1464506/Hallitusohjelma_27052015_12998.pdf/ae088a77-b0ab-4964-846d-1e7d14a9d064
Luettu 15.10.2015.

Soikkeli, T-M. 2015 NoSQL-tietokannat: vertailu relaatiotietokantoihin ja luokittelu tietomallin sekä käyttökohteiden mukaan. Kandidaatintutkielma. Jyväskylän yliopisto, Tietojärjestelmätiede.

Spoiala C. 2015. Cloud offering: Comparison between IaaS, PaaS, SaaS, BaaS. Blogikirjoitus . Luettavissa: http://assist-software.net/blog/cloud-offering-comparison-between-iaas-paas-saas-baas. Luettu 9.11.2015.

Stonebraker, M. & Cattell, R. 2011. 10 rules for scalable performance in 'simple operation' datastores. Communications of the ACM, 54, 6 (June 2011), s. 72-80.

Strauch, C.& Kriha, W. (lecturer) 2011. NoSQL databases. Lecture Notes, Stuttgart Media University. Luettavissa: http://www.christof-strauch.de/nosqldbs.pdf. Luettu 1.9.2015.

Temmes, M. 2006. Valtionhallinto – jatkuvuutta ja muutosta. Suomen poliittinen järjestelmä, Luku 3.8 Valtionhallinto. Verkkokirja. Helsingin yliopisto. Luettavissa: http://blogs.helsinki.fi/vol-spj/valtionhallinto/jatkuvuutta-ja-muutosta/
Luettu 12.9.2015.

Tietohallintolaki. Laki julkisen hallinnon tietohallinnosta. 10.6.2011/634.

Vaasan yliopisto. Tietojenkäsittely TiTe.1020. Tietokanta. Luettavissa: http://slideplayer.biz/slide/1981864/. Luettu 2.9.2015.

Vainio, J. 2012. XILtoSQL – hierarkkisten kyselyiden semantiikka relaatiotietokannassa. Pro gradu –tutkielma. Tampereen yliopisto, Informaatiotieteiden yksikkö.

Van Der Lans, R. F. 2015. Big Data is not always Big Information. Datavirtualization-blogikirjoitus. Luettavissa: http://www.datavirtualizationblog.com/big-data-is-not-always-big-

informati-
on/?utm_source=Blog&utm_medium=SM&utm_term=RVDL&utm_campaign=Blog-SM-RVDL. Luettu 14.11.2015.

Verohallinnon päätös yleisestä tiedonantovelvollisuudesta. Antopäivä: 31.12.2014. Diaarinumero: A182/200/2014. Luettavissa: http://www.vero.fi/fi-FI/Syventavat_veroohjeet/Verohallinnon_paatokset/Verohallinnon_paatos_yleisesta_tiedonant(35374). Luettu 1.8.2015.

Verohallinnon strategia 2013–2018. Luettavissa: http://www.vero.fi/fi-FI/Tietoa_Verohallinnosta/Verohallinto/Verohallinnon_esittely/Verohallinnon_strategia_20132018(12997). Luettu 1.8.2015.

Verohallinnon tilinpäätös 2014. Luettavissa: http://www.vero.fi/download/Verohallinnon_tilinpaatos_2014/%7B03F12DB7-B08A-49DA-B74F-5ABC7EE52D30%7D/10635. Luettu 1.8.2015.

Verohallinnon tilinpäätös 2006. Luettavissa: http://www.vero.fi/download/Verohallinnon_tilinpaatos_2006/%7BB53A3C79-784D-48CA-9E7C-2F06BD287891%7D/6490. Luettu 1.8.2015.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010). A comparison of a graph database and a relational database: a data provenance perspective. Teoksessa Proceedings of the 48th annual southeast regional conference (s. 42)

Zhang Q., Cheng L. & Boutaba R. 2010. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 2010, 1.1: 7-18.